BRL MR 2712

BRL

12

NW

MEMORANDUM REPORT NO. 2712

GEOMETRY PROGRAMS TO AID IN PRODUCING COMBINATORIAL GEOMETRY TARGET DESCRIPTIONS

Keith Applin
Gary Kuehl

December 1976

USA BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: BRL Memorandum Report No. 2712

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): GEOMETRY PROGRAMS TO AID IN PRODUCING COMBINATORIAL GEOMETRY TARGET DESCRIPTIONS

5. TYPE OF REPORT & PERIOD COVERED: FINAL rept.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Keith Applin, Gary Kuehl

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS: USA Ballistic Research Laboratory, Aberdeen Proving Ground, MD 21005

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1L662618AH80

11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Materiel Development & Readiness Command, 5001 Eisenhower Ave, Alexandria, VA 22333

12. REPORT DATE: DECEMBER 1976

13. NUMBER OF PAGES: 106

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): BRL-MR-2712

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GIFT	mini-computer programs
COM-GEOM	BASIC programs
Combinatorial Geometry	target description data
Wang 2200	target geometry data
Geometrical programs	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(DTSebold)

These programs were written for the WANG 2200 mini-computer system to increase the accuracy and reduce the time needed to develop COM-GEOM target descriptions. These programs were written for cases where quick turn-around time is imperative.

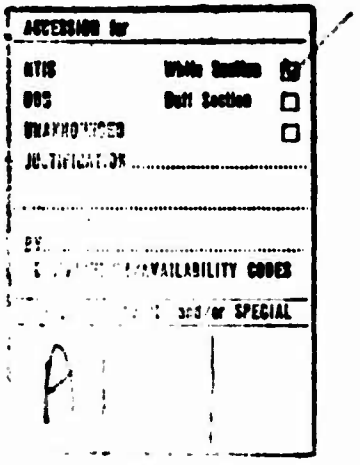
This report contains a discussion, sample runs and a listing (BASIC language) for each program.

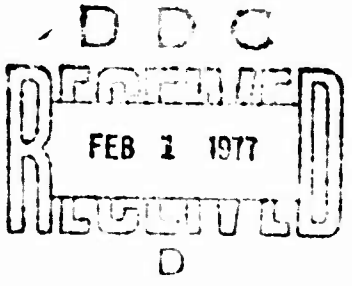
DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE 050750

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

TABLE OF CONTENTS





I. INTRODUCTION

Most vulnerability lethality analyses completed at the Ballistic Research Laboratory (BRL) today require some type of geometric description of the target. This geometric description is usually completed employing the combinatorial geometry (COM-GEOM) technique. The COM-GEOM technique of target description represents components of a target as Boolean combinations of twelve basic geometric solids. These solids are listed in Table I.

As part of a continuing effort to increase the accuracy and reduce the production time of these COM-GEOM descriptions, a set of geometry aid programs has been developed. The purpose of these aid programs is to quickly give the COM-GEOMer solutions to the bulk of geometry problems that arise during the production of a target description.

Since quick turnaround time is imperative, these aid programs have been coded for the WANG 2200 mini-computer system. Many of these programs are also coded and available for the WANG 700 series calculator. The BASIC type language used on the WANG 2200 system can be converted to other systems, if necessary.

This report documents the aid programs which are presently available for the WANG 2200 mini-computer system.

II. BACKGROUND

In the course of producing COM-GEOM target descriptions, one soon becomes aware of the urgent need for solutions to relatively simple geometric problems. These include such problems as vectors normal to other vectors; four points in a plane; the intersection point of two lines, etc. The requirement for quick, accurate answers to such problems led to the development of this set of mini-computer codes. Indexed in Table II are the names and functions of each code. These available codes will be presented in a user oriented manner. For each of the geometry aid programs, the following are included:

(1) Descriptive paragraphs on the function and use
(2) Memory requirement
(3) Any restrictions, limitations, or special features
(4) Instructions for use on the WANG 2200 system
(5) Sample output listings (note: whenever feasible at least one of the sample outputs will be a case where the answers can be readily checked)

Appendix A contains listings of the codes for each program. All these codes are available at the BRL on cassette magnetic tape.

Table I. Geometric Solids Used In COM-GEOM Descriptions

SYMBOL	SOLID NAME
RPP	Rectangular Parallepiped
BOX	Box
RAW	Right Angle Wedge
ARB	Arbitrary Convex Polyhedron
ARS	Triangular Surfaced Polyhedron
ELL	Ellipsoid of Revolution
SPH	Sphere
RCC	Right Circular Cylinder
REC	Right Elliptical Cylinder
TRC	Truncated Right Angle Cone
TEC	Truncated Elliptic Cone
TOR	Torus

Table II. Index of the Geometry Aid Programs

Number	Name	Purpose
1	SPHERE	Finds a sphere defined by 4 nonplanar points on the surface of the sphere.
2	CIRCIR	Finds the intersection points of 2 circles (2 dimensions).
3	CIRCLE	Finds a circle defined by 3 noncollinear points on the circumference of the circle (2 dimensions).
4	RCC	Finds RCC defined by 3 noncollinear points on circumference of the base and a desired height. Can also be used in same manner as #3 but in 3 dimensions.
5	LINECIR	Finds the intersection points of a line and circle (2 dimensions).
6	TANCIR	Finds tangent points on a circle from a point outside the circle (2 dimensions).
7	PLANEINT	Finds the point of intersection of 3 different planes.
8	LINEPLAN	Finds the point of intersection of a line and plane.
9	LINELINE	Finds intersection point of 2 lines (2 dimensions).
10	RFARB	Finds an ARB8 defined by (1) point on one face; (2) rotation and fallback angles of that face; (3) 2 coordinates of the remaining 3 points on that face, and (4) a desired thickness.
11	3PTARB	Finds an ARB8 defined by (1) 3 noncollinear points on one face; (2) 2 coordinates of the remaining point on that face; (3) a desired thickness.
12	NORMVEC	Finds (1) vector of desired length in the direction of a given vector and (2) a vector of desired length perpendicular to the given vector (2 dimensions).
13	PERPENV	Finds (1) a vector of desired length perpendicular to 2 given vectors and (2) the angle between the 2 given vectors.
14	AMTRACK	Adds tracks to domestic vehicles - listing the solids and a region table.

Table II. Index of the Geometry Aid Programs (Continued)

Number	Name	Purpose
15	SOLIDROT	Rotates selected COM-GEOM solids about any point in the XY, XZ, or YZ planes.
16	PLOTSOL	Plots selected COM-GEOM solids at any desired aspect. Nc hidden line algorithm.
17	DARBIN	Corrects a "bad" ARB and computes an inside one.
18	BOXIN	Corrects a "bad" BOX and computes an inside one.
19	RAWIN	Corrects a "bad" RAW and computes an inside one.
20	TRCIN	Computes an inside TRC.
21	RECIN	Corrects a "bad" REC and computes an inside one.
22	TECIN	Corrects a "bad" TEC and computes an inside one.
23	PARB	Computes the points of intersect of a set of planes taken 3 at a time.

All of these programs have been coded to interact with the user via the scope of the WANG 2200 system. Each input is asked for in a step-wise manner and all instructions are listed on the screen. The loading instructions for these programs are all the same.

To load any of these programs from cassette tape:

(1) Place the tape in the machine and rewind
(2) Key LOAD "Program Name" CR/LF
(3) After program is loaded in machine, key RUN CR/LF.

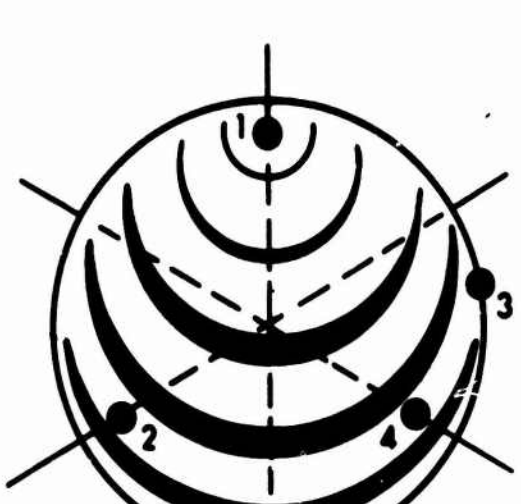
At this point instructions and the request for the first inputs will be displayed on the screen. For specific directions on the input required see the instruction section of the individual program. When finished running any of the programs perform the following steps:

(1) Key RESET
(2) Key CLEAR CR/LF
(3) Rewind the tape and remove it from the machine.

Even though some of these programs operate on two-dimensional problems only, they are useful because many three-dimensional space problems can be reduced to the two-dimensional case. For example, most of the boxes (BOX) used in COM-GEOM descriptions employ vectors with only 2 nonzero components and many right circular cylinders (RCC) have height vectors of 2 (and often only 1) nonzero components.

III. DISCUSSION OF THE PROGRAMS

1. **Program Name:** SPHERE
   Description: This program finds the sphere defined by any four noncoplanar points on the surface of the sphere.



Memory: 1467 bytes
Restrictions: None
Instructions: After loading the program, enter the following: (enter data separated by commas)

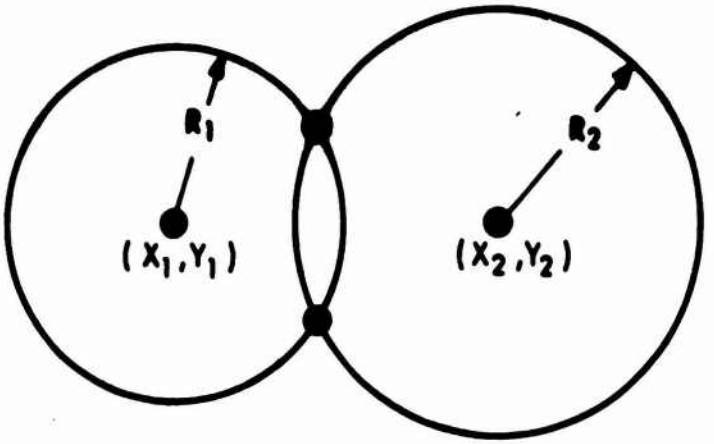
(a) X,Y,Z coordinates of point 1 on the surface CR/LF
(b) X,Y,Z coordinates of point 2 on the surface CR/LF
(c) X,Y,Z coordinates of point 3 on the surface CR/LF
(d) X,Y,Z coordinates of point 4 on the surface CR/LF
(e) The results will be printed out
(f) To run again key CONTINUE CR/LF and program returns to step (a).

Sample Outputs: First sample run is case which is readily checked.

```
INPUT POINTS
PT          X              Y              Z
 1        5.0000         0.0000         0.0000
 2        0.0000         5.0000         0.0000
 3        0.0000         0.0000         5.0000
 4        0.0000         0.0000        -5.0000
SPHERE    XC=      0.0000   YC=      0.0000   ZC=      0.0000
          RADIUS=      5.0000
```

```
INPUT POINTS
PT          X                Y                Z
 1       253.2300          65.2340       -1025.3260
 2       123.5402         321.2548         321.0214
 3         1.0000           2.0000           3.0000
 4         8.0000           6.0000           1.0000
SPHERE     XC=   3727.2686   YC=  -1289.6832   ZC=    270.7310
           RADIUS=  3947.7220
```

2. Program Name: CIRCIR
Description: This program finds the intersection points of 2 circles (in 2 dimensions). This program is useful in checking for the overlapping of stored missiles or ammunition.



Memory: 1502 bytes
Restrictions: None
Instructions: After loading the program, enter the following: (enter data separated by commas)
(a) The X,Y coordinates of the center and the radius of circle 1 CR/LF
(b) The X,Y coordinates of the center and the radius of circle 2 CR/LF
(c) Results will be printed out
(d) To run again key CONTINUE CR/LF and program returns to step (a).

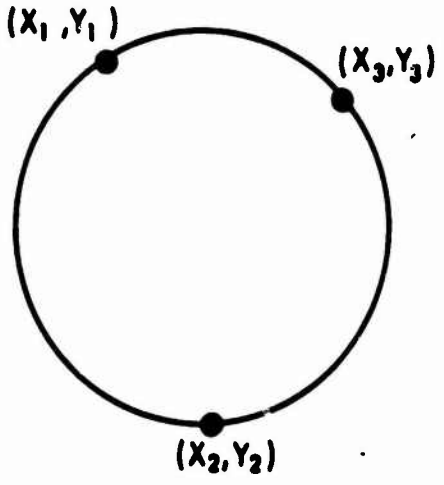
Sample Outputs: First sample output is a case where circles are tangent. The second sample run is a case where one circle is inside the other. Note that the program gives a "no intersection" result.

```
CIRCLE 1    X CENT=     0.0000   Y CENT=     0.0000   R=     5.0000
CIRCLE 2               15.0000              0.0000          10.0000
CIRCLES TANGENT AT     X=     5.0000    Y=     0.0000
```

CIRCLE 1	X CENT=	0.0000	Y CENT=	0.0000	R=	5.0000
CIRCLE 2		0.0000		0.0000		10.0000
NO INTERSECTION						

CIRCLE 1	X CENT=	12.3564	Y CENT=	-36.6523	R=	59.2600
CIRCLE 2		15.2360		39.2365		56.3210
INTERSECTION POINTS		X= 57.3818		Y= 1.8763		
		-29.6198		5.1776		

3. Program Name: CIRCLE
Description: This program finds the circle defined by 3 noncollinear points on the circumference of the circle (in 2 dimensions). If the 3 dimensional case is required use program 4 (RCC).

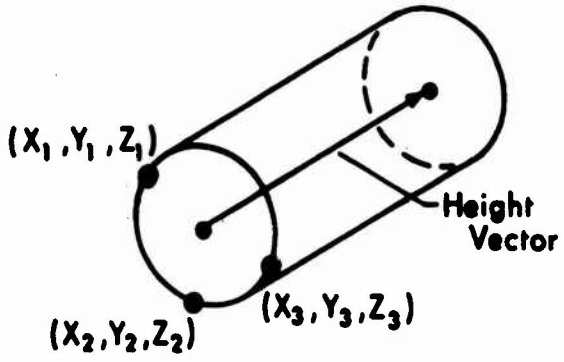


Memory: 1093 bytes
Restrictions: None
Instructions: After loading the program, enter the following: (enter data separated by commas)
(a) The 2 coordinates of point 1 CR/LF
(b) The 2 coordinates of point 2 CR/LF
(c) The 2 coordinates of point 3 CR/LF
(d) Results will be printed out
(e) To run another case, key CONTINUE CR/LF and program returns to step (a).

Sample Outputs:

```
          INPUT POINTS
            X              Y
         5.0000         0.0000
         0.0000         5.0000
         0.0000        -5.0000
XCENT=     0.0000   YCENT=     0.0000   RAD=     5.0000

          INPUT POINTS
            X              Y
       125.3650       613.2483
      -561.2357      6512.3570
       -98.2350       251.2478
XCENT= -4287.7987   YCENT=  3089.1008   RAD=  5060.2272
```

4. Program Name: RCC
Description: This program finds an RCC defined by: (1) 3 noncollinear points on the circumference of the base; (2) length of height vector desired.



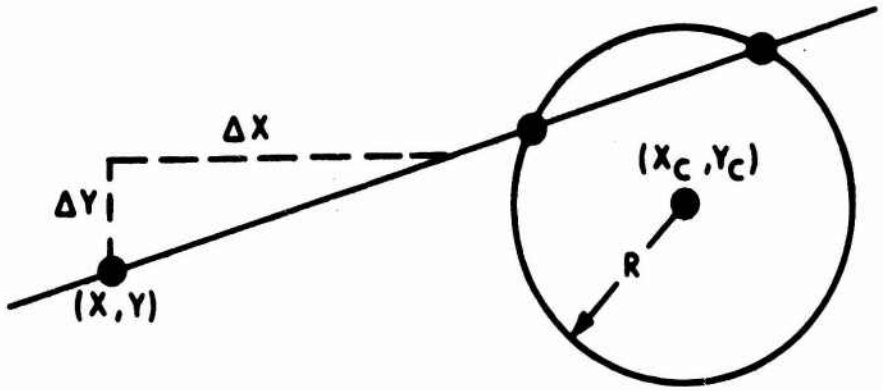
Memory: 3646 bytes
Restrictions: The user must beware of the direction of the height vector. The program will print out the two possible height vectors and the user must choose the one he desires. Note that this program can also be used to fit a circle (in three dimensions) through 3 noncollinear points.
Instructions: After loading the program, enter the following: (enter data separated by commas)
(a) Key X,Y,Z coordinates of point 1 on the circumference CR/LF
(b) Key X,Y,Z coordinates of point 2 on the circumference CR/LF
(c) Key X,Y,Z coordinates of point 3 on the circumference CR/LF
(d) Key the length of height vector desired CR/LF
(e) Results will be printed out
(f) Program returns to step (a).

Sample Outputs: Note that if this program is used to fit a circle through 3 points, use the circle defined by the center and radius of the base of the RCC.

```
INPUT POINTS:
POINT 1    X=    10.0000    Y=     0.0000    Z=     0.0000
POINT 2    X=     0.0000    Y=    10.0000    Z=     0.0000
POINT 3    X=   -10.0000    Y=     0.0000    Z=     0.0000
LENGTH OF HEIGHT VECTOR=          50

THE PARAMETERS OF THE RCC:
CENTER OF BASE  XC=     0.0000 YC=     0.0000 ZC=     0.0000
HEIGHT VECTOR   DX=     0.0000 DY=     0.0000 DZ=   -50.0000
      OR        DX=     0.0000 DY=     0.0000 DZ=    50.0000
RADIUS OF BASE=    10.0000
```

```
INPUT POINTS:
POINT 1    X=   542.3265   Y=   321.1250   Z=  8235.3250
POINT 2    X=   586.3241   Y=   652.1247   Z=   -74.2541
POINT 3    X=   821.0214   Y=  -652.3214   Z=   862.3214
LENGTH OF HEIGHT VECTOR=         100

THE PARAMETERS OF THE RCC:
CENTER OF BASE  XC=   161.9633 YC=  2608.4520 ZC=  4162.9243
HEIGHT VECTOR   DX=    98.2502 DY=    18.5822 DZ=     1.2604
      OR        DX=   -98.2502 DY=   -18.5822 DZ=    -1.2604
RADIUS OF BASE=  4686.2548
```

5. Program Name: LINECIR
Description: This program finds the intersection points of a line and a circle (in 2 dimensions). The line is entered using the point slope method.



Memory: 1559 bytes
Restrictions: None
Instructions: After the program is loaded, enter the following: (enter the data separated by commas)
(a) X and Y coordinates, DELTA X, DELTA Y, for the line CR/LF
(b) X,Y coordinates of center and radius of circle CR/LF
(c) Results will then be printed out
(d) To run another case, key CONTINUE CR/LF and program will return to step (a).

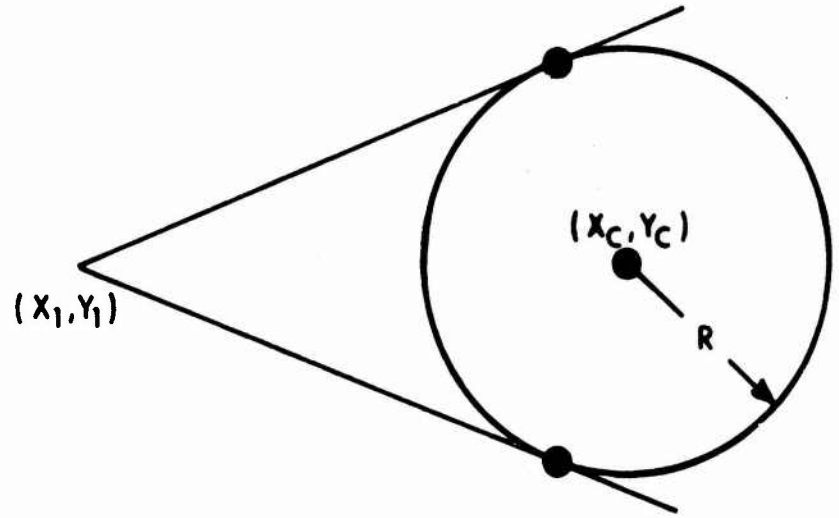
Sample Outputs:

```
LINE INPUT        X=      0.0000        Y=      0.0000
                DEL X=      1.0000    DEL Y=      1.0000
CIRCLE INPUT    XC=       0.0000   YC=       0.0000   R=       1.0000
INTERSECTION POINTS
X=      0.7071    Y=      0.7071
X=     -0.7071    Y=     -0.7071
```

```
LINE INPUT        X=      5.0000        Y=     10.0000
                DEL X=      0.0000    DEL Y=     11.0000
CIRCLE INPUT    XC=       0.0000   YC=       0.0000   R=       1.0000
NO INTERSECTION
```

```
LINE INPUT        X=      0.0000        Y=      0.0000
                DEL X=      0.0000   DEL Y=     11.0000
CIRCLE INPUT    XC=      0.0000  YC=      0.0000  R=      5.0000
INTERSECTION POINTS
X=      0.0000    Y=      5.0000
X=      0.0000    Y=     -5.0000


LINE INPUT        X=      5.0000        Y=     10.0000
                DEL X=      0.0000   DEL Y=     11.0000
CIRCLE INPUT    XC=      0.0000  YC=      0.0000  R=      5.0000
LINE TANGENT TO CIRCLE AT   X=      5.0000   Y=      0.0000


LINE INPUT        X=   1203.2564        Y= -2301.2356
                DEL X=   1267.2175   DEL Y= -3691.2547
CIRCLE INPUT    XC=    235.6540  YC=  -326.1250  R=   3214.2560
INTERSECTION POINTS
X=   1534.5473    Y= -3266.2470
X=   -545.2092    Y=  2791.8381
```

6. Program Name: TANCIR
Description: This program finds the tangent points on a circle from a point outside the circle. (2 dimensions)



Memory: 1153 bytes
Restrictions: None
Instructions: After the program is loaded, enter the following: (enter data separated by commas)
(a) X and Y coordinates of the point outside the circle CR/LF
(b) X and Y coordinates of the center of the circle and the radius of the circle CR/LF
(c) Results will be printed out
(d) To run another case, key CONTINUE CR/LF and program returns to step (a).

Sample Outputs:

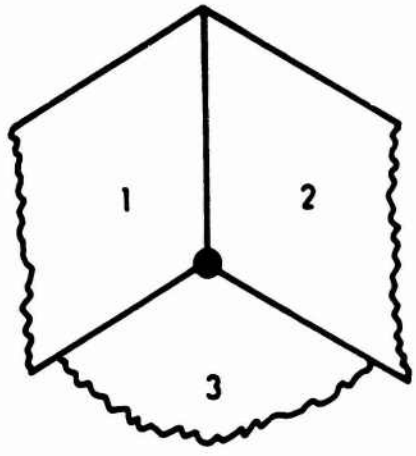
```
POINT INPUT  X=     0.0000  Y=     0.0000
CIRCLE INPUT XCENT=     5.0000  YCENT=     2.0000  R=     2.0000
TANGENT PTS  X=     5.0000  Y=     0.0000
             X=     3.6206  Y=     3.4482
```

```
POINT INPUT  X=    50.0000  Y=     0.0000
CIRCLE INPUT XCENT=     0.0000  YCENT=     0.0000  R=    10.0000
TANGENT PTS  X=     2.0000  Y=     9.7979
             X=     2.0000  Y=    -9.7979
```

```
POINT INPUT  X=    35.0000  Y=     0.0000
CIRCLE INPUT  XCENT=     0.0000  YCENT=     0.0000  R=    35.0000
POINT IS ON CIRCLE


POINT INPUT  X=     0.0000  Y=     0.0000
CIRCLE INPUT  XCENT=     0.0000  YCENT=     0.0000  R=    50.0000
POINT IS INSIDE CIRCLE



POINT INPUT  X=  1234.5678  Y=  2345.6789
CIRCLE INPUT  XCENT=   120.3650  YCENT=  3125.0140  R=   382.3600
TANGENT PTS   X=   418.7842  Y=  3364.0644
              X=    -1.8377  Y=  2762.7079


POINT INPUT  X=   102.3640  Y=  -326.2589
CIRCLE INPUT  XCENT=     0.0000  YCENT=     0.0000  R=    12.3000
TANGENT PTS   X=    11.8607  Y=     3.2576
              X=   -11.5953  Y=    -4.1019
```

7. Program Name: PLANEINT

Description: This program finds the point of intersection of three different planes. Each plane can be entered using one of three methods: (1) 3 noncollinear points; (2) one point on the plane and rotation (azimuth) and fallback (elevation) angles of that plane; or (3) coefficients of the equation of the plane - that is A,B,C,D of the equation AX + BY + CZ = D.



Memory: 3632 bytes

Restrictions: If the three planes intersect in a line the program will print out the message that the planes do not intersect.

Instructions: After the program is loaded, enter the following:

(a) An indicator depicting the method of input for the first plane where:

Indicator	Type of Plane Input
1	3 noncollinear points
2	Point; rotation, fallback angles
3	Coefficients of the plane

(b) Enter plane 1 by method chosen

(c) Next, enter the other 2 planes using steps (a) and (b) above

(d) Results will then be printed out

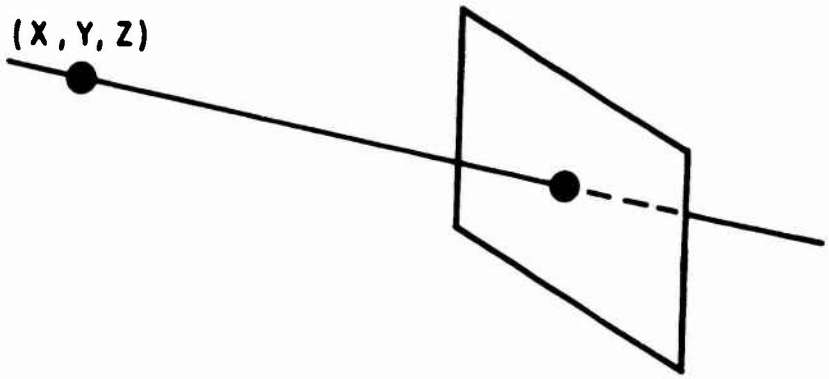
(e) To run another case, key CONTINUE CR/LF and program will return to step (a).

Sample Outputs: The first sample run is a case easily checked. In the second sample run all three methods of entering planes was used. Plane 1 was entered by the 3 point method, plane 2 by the point and rotation fallback method, and plane 3 by coefficients of the plane.

```
 INPUT FOR PLANE  1
COEFFICIENTS   A=      1.00000  B=      0.00000  C=      0.00000  D=     99.00000
INPUT FOR PLANE  2
COEFFICIENTS   A=      0.00000  B=      1.00000  C=      0.00000  D=     99.00000
INPUT FOR PLANE  3
COEFFICIENTS   A=      0.00000  B=      0.00000  C=      1.00000  D=     99.00000
INTERSECTION POINT   X=     99.0000  Y=     99.0000  Z=     99.0000


INPUT FOR PLANE  1
  X=  1263.3230  Y=   922.2570  Z=   320.0000  ROT=  84.933  F B=  61.673
      1063.2365      930.2451      325.2360
      1145.2368      787.0169      398.2360
COEFFICIENTS   A= 0.04190  B= 0.47263  C= 0.88025  D=   770.5159
INPUT FOR PLANE  2
  X=  1025.3690  Y=   812.3697  Z=   546.2387  ROT=  42.000  F B=  20.360
COEFFICIENTS   A= 0.69671  B= 0.62732  C= 0.34791  D=  1414.0592
INPUT FOR PLANE  3
COEFFICIENTS   A=     12.36980  B=     -3.26540  C=    210.36910  D=     56.23499
INTERSECTION POINT   X=    597.5406  Y=   1596.0651  Z=    -10.0938
```

8. Program Name: LINEPLAN

Description: This program finds the intersection point of a line and a plane. The line is entered using the point-slope method. The plane is entered one of three ways: (1) 3 noncollinear points; (2) point in the plane plus rotation (azimuth) angle and fallback (elevation) angle of the plane; or (3) coefficients A,B,C,D of the equation of the plane AX + BY + CZ = D.



Memory: 3360 bytes

Restrictions: None

Instructions: After the program is loaded, enter the following: (enter data separated by commas)

(a) X,Y,Z coordinates of a point on the line CR/LF

(b) DELTA X, DELTA Y, DELTA Z of the line CR/LF

(c) Indicator for type of plane input desired, where:

Indicator	Type of Plane Input
1	3 noncollinear points
2	Point, rotation angle, fallback angle
3	Coefficients of the plane equation

(d) Enter plane according to the method chosen

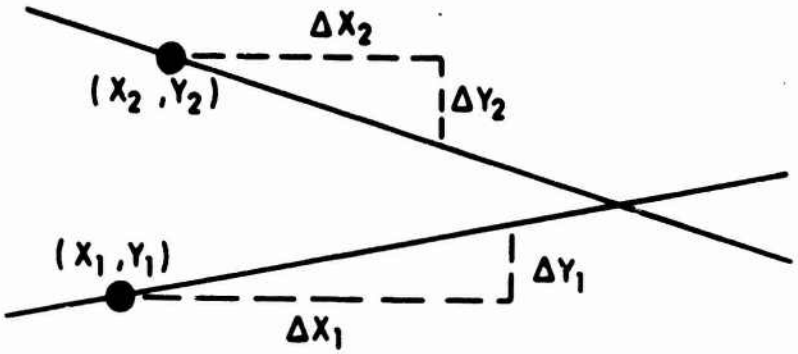
(e) Results will be printed out

(f) To run another case, key CONTINUE CR/LF and program will return to step (a).

Sample Outputs: The first three sample outputs are the same case but the plane has been entered using the three different methods.

```
INPUT FOR PLANE
COEFFICIENTS   A=     1.00000  B=     0.00000  C=     0.00000  D=   100.00000

LINE INPUT
X=     0.0000  Y=     0.0000  Z=     0.0000
DEL X=     2.0000  DEL Y=     2.0000  DEL Z=     2.0000

INTERSECTION POINT     X=   100.0000  Y=   100.0000  Z=   100.0000


INPUT FOR PLANE
  X=   100.0000  Y=     2.0000  Z=     3.0000  ROT=   0.000  F B=   0.000
COEFFICIENTS   A= 1.00000  B= 0.00000  C= 0.00000  D=   100.0000

LINE INPUT
X=     0.0000  Y=     0.0000  Z=     0.0000
DEL X=     2.0000  DEL Y=     2.0000  DEL Z=     2.0000

INTERSECTION POINT     X=   100.0000  Y=   100.0000  Z=   100.0000


INPUT FOR PLANE
  X=   100.0000  Y=     2.0000  Z=     3.0000  ROT=   0.000  F B=   0.000
       100.0000          5.0000          6.0000
       100.0000          6.0000          9.0000
COEFFICIENTS   A= 1.00000  B= 0.00000  C= 0.00000  D=   100.0000

LINE INPUT
X=     0.0000  Y=     0.0000  Z=     0.0000
DEL X=     2.0000  DEL Y=     2.0000  DEL Z=     2.0000

INTERSECTION POINT     X=   100.0000  Y=   100.0000  Z=   100.0000


INPUT FOR PLANE
  X=    23.5600  Y=   378.2360  Z=  -657.2350  ROT= 331.100  F B= -25.209
       120.3245       -23.1500       -65.2430
        92.1547     -3256.2140      3201.2450
COEFFICIENTS   A= 0.79203  B=-0.43724  C=-0.42593  D=   133.2206

LINE INPUT
X=   125.6000  Y=   256.3240  Z=  -235.6500
DEL X= -2354.0123  DEL Y=  3500.2345  DEL Z=    56.2340

INTERSECTION POINT     X=   156.5755  Y=   209.2440  Z=  -236.2065
```

9. Program Name: LINELINE
Description: This program finds the point of intersection of two lines (in 2 dimensions). The lines are entered using the point-slope method.



Memory: 1108 bytes
Restrictions: None
Instructions: After the program is loaded, enter the following:
(enter the data separated by commas)
(a) X and Y coordinates of a point on the line, DELTA X, DELTA Y for line 1 CR/LF
(b) X and Y coordinates of a point on the line, DELTA X, DELTA Y for line 2 CR/LF
(c) Results will be printed out
(d) To run another case, key CONTINUE CR/LF and program will return to step (a).

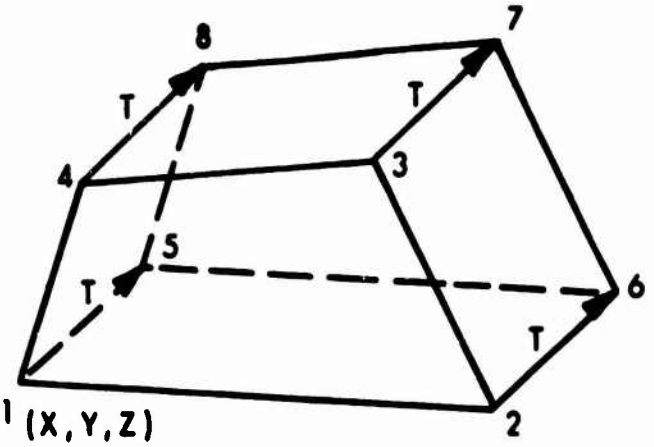
Sample Outputs:

```
LINE          X             Y          DEL X          DEL Y
  1        0.0000        0.0000         1.0000         1.0000
  2        2.0000        0.0000         0.0000        11.0000
INTERSECTION POINT    X=      2.0000  Y=      2.0000

LINE          X             Y          DEL X          DEL Y
  1        0.0000        0.0000         1.0000         1.0000
  2        2.0000        0.0000         1.0000         1.0000
LINES DO NOT INTERSECT

LINE          X             Y          DEL X          DEL Y
  1      120.3692      -23.5693      1235.6470      -321.2014
  2      104.2350      364.2580       -23.5697      1452.3699
INTERSECTION POINT    X=    110.4871  Y=    -21.0010
```

10. Program Name: RFARB

Description: This program finds an ARB8 with face 1234 parallel to face 5678 and a normal distance of any desired length (T) apart (see figure below). Face 1234 is defined by the following:

(1) one point on the face
(2) two coordinates of the remaining three points on that face
(3) rotation (azimuth) angle for that face
(4) fallback (elevation) angle for that face



Memory: 2006 bytes

Restrictions: Note that ARB6s can also be done by making point 4 equal to point 1. The user must beware that face 5678 is on the desired side of face 1234. If not, run the program again using a negative thickness T (see sample outputs 1 and 2).

Instructions: After the program has been loaded, enter the following:
(enter data separated by commas)

(a) The X,Y,Z coordinates of one point on face 1234 and the rotation and fallback angles of face 1234 CR/LF

(b) Two coordinates of the remaining 3 points will then be entered in the following manner:
For each point, you will be asked to enter indicators and 2 known coordinates. If the X and Y coordinates are known then enter 0,0,1,X coordinate, Y coordinate. Follow the same scheme for other coordinates. So enter the remaining 3 points as
1,0,0,Y coordinate, Z coordinate CR/LF
or 0,1,0,X coordinate, Z coordinate CR/LF
or 0,0,1,X coordinate, Y coordinate CR/LF

(c) The distance between face 1234 and face 5678 (this variable is called thickness) CR/LF

(d) Results will be printed out

(e) To run another case, key CONTINUE CR/LF and program will return to step (a). At this point it is usually advisable to run another case with a thickness which is the negative of the previous case's thickness.

Sample Outputs:

PT	X	Y	Z
1	1063.0000	922.2570	320.0000
2	1024.6265	859.0320	514.5840
3	1090.4640	787.1626	512.2850
4	1146.4420	787.1570	398.0010
5	1067.9665	926.7288	322.4324
6	1029.5930	863.5038	517.0164
7	1095.4305	791.6345	514.7174
8	1151.4085	791.6288	400.4334

FACE	A	B	C	D	ROT	F B
1234	0.69832	0.62877	0.34202	1431.66287	42.00	20.00
5678	0.69832	0.62877	0.34202	1438.77487	42.00	20.00

THICKNESS= 7.112

PT	X	Y	Z
1	1063.0000	922.2570	320.0000
2	1024.6265	859.0320	514.5840
3	1090.4640	787.1626	512.2850
4	1146.4420	787.1570	398.0010
5	1058.0334	917.7851	317.5675
6	1019.6600	854.5601	512.1515
7	1085.4974	782.6908	509.8525
8	1141.4754	782.6851	395.5685

FACE	A	B	C	D	ROT	F B
1234	0.69832	0.62877	0.34202	1431.66287	42.00	20.00
5678	0.69832	0.62877	0.34202	1424.55087	42.00	20.00

THICKNESS= -7.112

PT	X	Y	Z
1	123.2650	23.1200	55.0000
2	231.2560	-623.1200	55.0000
3	56.2350	-623.2150	55.0000
4	532.3690	-458.2147	55.0000
5	123.2650	23.1200	75.0000
6	231.2560	-623.1200	75.0000
7	56.2350	-623.2150	75.0000
8	532.3690	-458.2147	75.0000

FACE	A	B	C	D	ROT	F B
1234	0.00000	0.00000	1.00000	55.00000	0.00	00.00
5678	0.00000	0.00000	1.00000	75.00000	0.00	00.00

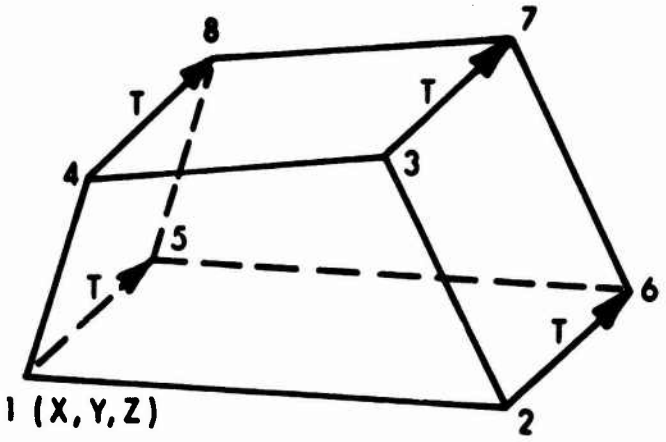
THICKNESS= 20

11. Program Name: 3PTARB

Description: This program finds an ARB8 with the face 1234 parallel to the face 5678 and a normal distance of any desired length (T) apart (see figure below). Face 1234 is defined by the following:

(1) 3 noncollinear points and
(2) 2 coordinates of the remaining point

This program is useful in finding armor plates of any desired thickness.



Memory: 2666 bytes

Restrictions: ARB6s can be done by making the 4th point equal to point 1. The user must beware that 5678 is on the correct side of face 1234. If not, rerun the program using a negative T.

Instructions: After the program has been loaded, enter the following:
(enter the data separated by commas)

(a) The X,Y,Z coordinates of point 1 CR/LF
(b) The X,Y,Z coordinates of point 2 CR/LF
(c) The X,Y,Z coordinates of point 3 CR/LF
(d) The remaining point of face 1234 in the same manner as the RFARB program, that is:
    1,0,0,Y coordinate, Z coordinate CR/LF
    or 0,1,0,X coordinate, Z coordinate CR/LF
    or 0,0,1,X coordinate, Y coordinate CR/LF
(e) The thickness desired (i.e. the distance between face 1234 and face 5678)
(f) Results will be printed out
(g) To run another case, key CONTINUE CR/LF and the program will return to step (a). At this point, it is advisable to run the same problem only with the thickness negative to the thickness of the previous run.

Sample Outputs:

PT	X	Y	Z
1	1063.0000	922.2570	320.0000
2	1024.6235	859.0320	514.5840
3	1090.4640	787.1626	512.3350
4	1146.4420	737.1570	398.0010
5	1067.9665	926.7238	322.4324
6	1029.5930	863.5038	517.0164
7	1095.4305	791.6344	514.7174
8	1151.4085	791.6288	400.4334

FACE	A	B	C	D	ROT	FB
1234	0.69832	0.62877	0.34202	1431.66231	42.00	20.00
5678	0.69832	0.62877	0.34202	1438.77461	42.00	20.00

THICKNESS= 7.112

PT	X	Y	Z
1	99.0000	123.0000	456.0000
2	99.0000	450.0000	784.0000
3	99.0000	569.0000	912.0000
4	99.0000	5632.0000	8952.0000
5	109.0000	123.0000	456.0000
6	109.0000	450.0000	784.0000
7	109.0000	569.0000	912.0000
8	109.0000	5632.0000	8952.0000

FACE	A	B	C	D	ROT	FB
1234	1.00000	0.00000	0.00000	99.00000	0.00	0.00
5678	1.00000	0.00000	0.00000	109.00000	0.00	0.00

THICKNESS= 10

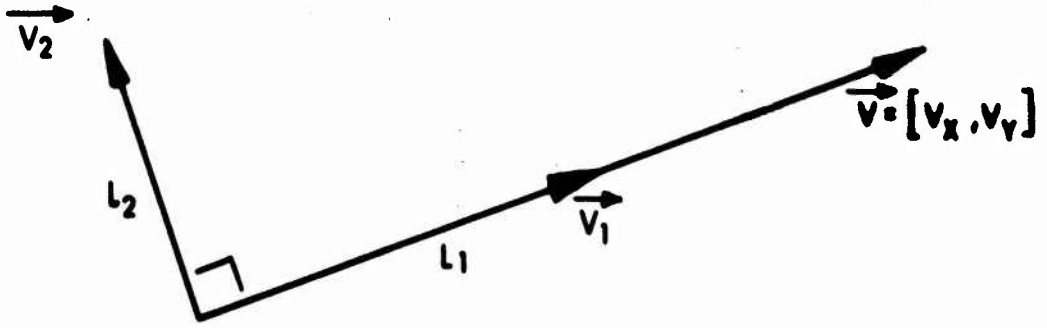
PT	X	Y	Z
1	23.0000	64.0000	100.0000
2	154.0000	237.0000	100.0000
3	564.0000	278.0000	100.0000
4	1254.0000	6587.0000	100.0000
5	23.0000	64.0000	110.0000
6	154.0000	237.0000	110.0000
7	564.0000	278.0000	110.0000
8	1254.0000	6587.0000	110.0000

FACE	A	B	C	D	ROT	FB
1234	0.00000	0.00000	-1.00000	-100.00000	*******	-90.00
5678	0.00000	0.00000	-1.00000	-110.00000	*******	-90.00

*** NOTE SINCE FB IS -90 ROT IS NOT UNIQUE***

THICKNESS= -10

12. Program Name: NORMVEC
Description: This program finds (1) a vector of any desired length in the same direction as a given vector, and (2) a vector of any desired length perpendicular to the given vector (2 dimensions). For a 3 dimensional case, use program 13, PERPENV.



Memory: 802 bytes
Restrictions: User must beware the direction of the perpendicular vector. If the one printed out is not the one desired, use its "negative" or run the problem again with negative length (see sample outputs 3 and 4).
Instructions: After the program is loaded, enter the following: (enter the data separated by commas)
(a) X and Y components of the given vector, length of vector in direction of given vector, length of perpendicular vector CR/LF
(b) Results will be printed out
(c) To run another case, key CONTINUE CR/LF and program will return to step (a).

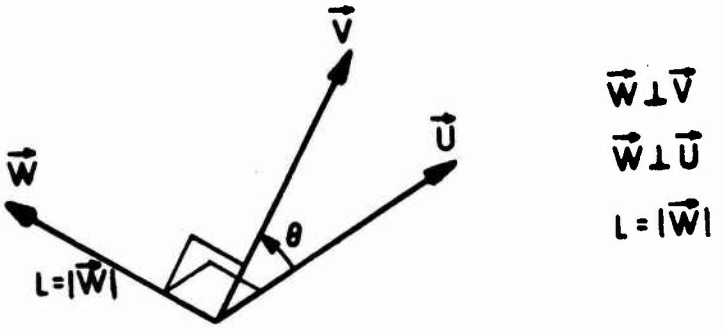
Sample Outputs:

```
INPUT VECTOR:
LENGTH=   100.000  DEL X=     0.000  DEL Y=   100.000
VECTOR IN DIRECTION OF INPUT VECTOR:
LENGTH=    10.000  DEL X=     0.000  DEL Y=    10.000
VECTOR PERPENDICULAR TO INPUT VECTOR:
LENGTH=    25.000  DEL X=   -25.000  DEL Y=     0.000


INPUT VECTOR:
LENGTH=   141.421  DEL X=   100.000  DEL Y=   100.000
VECTOR IN DIRECTION OF INPUT VECTOR:
LENGTH=    10.000  DEL X=     7.071  DEL Y=     7.071
VECTOR PERPENDICULAR TO INPUT VECTOR:
LENGTH=    25.000  DEL X=   -17.677  DEL Y=    17.677
```

```
INPUT VECTOR:
LENGTH=   261.178  DEL X=   123.456  DEL Y=  -230.157
VECTOR IN DIRECTION OF INPUT VECTOR:
LENGTH=    12.350  DEL X=     5.837  DEL Y=   -10.883
VECTOR PERPENDICULAR TO INPUT VECTOR:
LENGTH=    36.200  DEL X=    31.900  DEL Y=    17.111


INPUT VECTOR:
LENGTH=   261.178  DEL X=   123.456  DEL Y=  -230.157
VECTOR IN DIRECTION OF INPUT VECTOR:
LENGTH=    12.350  DEL X=     5.837  DEL Y=   -10.883
VECTOR PERPENDICULAR TO INPUT VECTOR:
LENGTH=   -36.200  DEL X=   -31.900  DEL Y=   -17.111
```

13. Program Name: PERPENV

Description: This program finds (1) a vector of any desired length perpendicular to two given vectors and (2) the angle between the two given vectors.



Memory: 1373 bytes

Restrictions: User must beware of the direction of the perpendicular vector. The program prints out both of the perpendicular vectors and the user must choose the one desired.

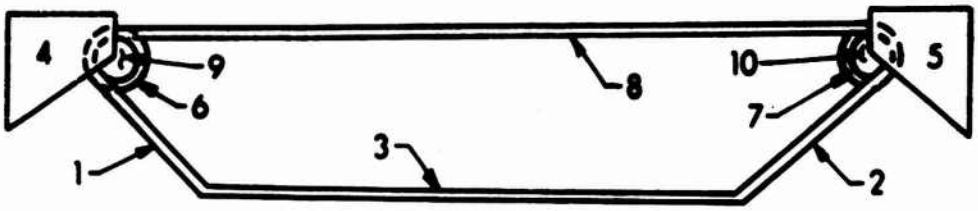
Instructions: After the program is loaded, enter the following: (enter the data separated by commas)

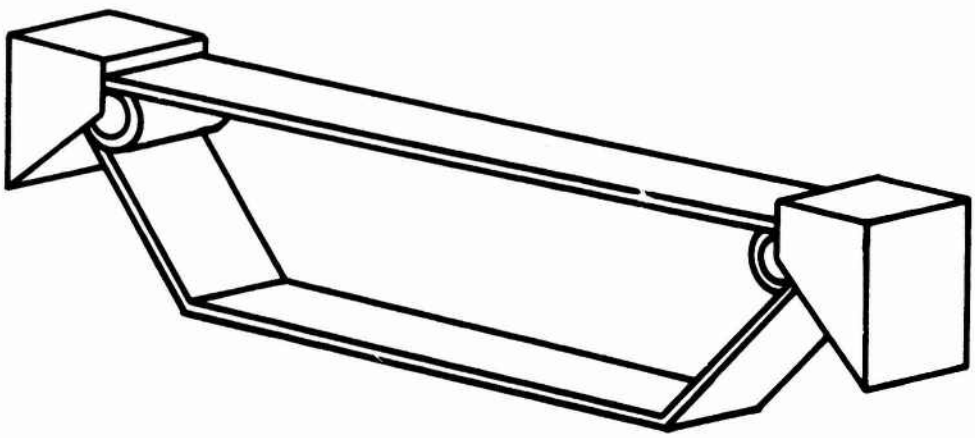
(a) The X,Y and Z components of vector 1 CR/LF
(b) The X,Y and Z components of vector 2 CR/LF
(c) The desired length of the perpendicular vector CR/LF
(d) Results will be printed out
(e) Program returns to step (a)

Sample Outputs:

```
INPUT VECTORS:
DEL X=    15.0000   DEL Y=     0.0000   DEL Z=     0.0000
DEL X=     0.0000   DEL Y=    15.0000   DEL Z=     0.0000
LENGTHS     VECTOR 1=   15.0000   VECTOR 2=   15.0000
ANGLE BETWEEN INPUT VECTORS=  90.000 DEGREES

VECTOR PERPENDICULAR TO INPUT VECTORS:
     DX=     0.0000   DY=     0.0000   DZ=    25.0000
 OR  DX=     0.0000   DY=     0.0000   DZ=   -25.0000



INPUT VECTORS:
DEL X=   156.2340   DEL Y=   -62.3147   DEL Z=   -91.2400
DEL X=   -61.2357   DEL Y=   -35.3214   DEL Z=   302.3540
LENGTHS     VECTOR 1=  191.3554   VECTOR 2=  305.1497
ANGLE BETWEEN INPUT VECTORS= 121.219 DEGREES

VECTOR PERPENDICULAR TO INPUT VECTORS:
     DX=   -60.2406   DY=  -137.0390   DZ=    -9.5581
 OR  DX=    60.2406   DY=   137.0390   DZ=     9.5581
```

14. Program Name: AMTRACK
Description: This program adds tracks to domestic vehicles, or any vehicle with "live" tracks. The two figures below are drawings of the track the program produces. Included in the output are the solid table and the region combination table.





Memory: 7497 bytes
Restrictions: The following assumptions are made:
(1) The Z-coordinates of all the road wheels are equal
(2) The "idler" wheel is to the front of the first road wheel
(3) The "drive" wheel is to the rear of the last road wheel
(4) The radii of the road wheels are equal
Instructions: After the program is loaded, enter the following:
(enter the data separated by commas)
(a) The X,Z coordinates of the first road wheel CR/LF
(b) The X,Z coordinates of the last road wheel CR/LF
(c) The radius of the road wheels CR/LF
(d) The X,Z coordinates of the "idler" wheel CR/LF

(e) The radius of the "idler" wheel CR/LF
(f) The X,Z coordinates of the "drive" wheel CR/LF
(g) The radius of the "drive" wheel CR/LF
(h) The Y min and Y max of the tracks CR/LF
(i) The thickness wanted for the tracks CR/LF
(j) Results will be printed out
(k) Program returns to step (a).

Sample Output: The following input data were used to produce this sample output (according to the order of input required in the instructions above:)

(1) 1500,100
(2) -1600,100
(3) 80
(4) 2000,500
(5) 60
(6) -2200,600
(7) 75
(8) 0,100
(9) 20

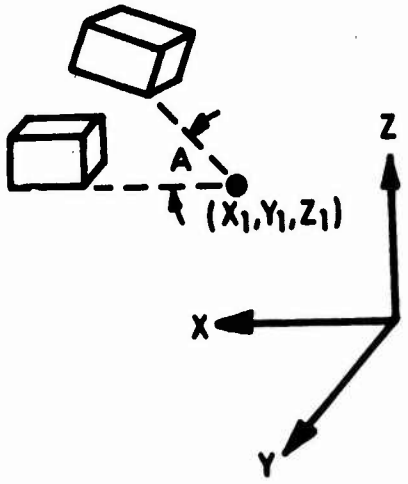
1 BOX	1539.3963	0.0000	0.0000	512.6554	0.0000	439.2495	FRNT SLOPE
	-13.0129	0.0000	15.1876	0.0000	100.0000	0.0000	
2 BOX	-1639.3383	0.0000	0.0000	-622.1040	0.0000	527.5443	REAR SLOPE
	12.9352	0.0000	15.2538	0.0000	100.0000	0.0000	
3 RPP	-1639.3383	1539.3963	0.0000	100.0000	0.0000	20.0000	TRACK BOT
4 ARB8	2247.2457	110.0000	630.0000	2247.2457	110.0000	211.4353	IDLER DUM
	2000.0000	110.0000	500.0000	2000.0000	110.0000	630.0000	
	2247.2457	-10.0000	630.0000	2247.2457	-10.0000	211.4353	
	2000.0000	-10.0000	500.0000	2000.0000	-10.0000	630.0000	
5 ARB8	-2455.4711	110.0000	745.0000	-2455.4711	110.0000	290.7368	DRIVE DUM
	-2200.0000	110.0000	600.0000	-2200.0000	110.0000	745.0000	
	-2455.4711	-10.0000	745.0000	-2455.4711	-10.0000	290.7368	
	-2200.0000	-10.0000	600.0000	-2200.0000	-10.0000	745.0000	
6 RCC	2000.0000	0.0000	500.0000	0.0000	100.0000	0.0000	TRCK IDLER
	80.0000						
7 RCC	-2200.0000	0.0000	600.0000	0.0000	100.0000	0.0000	TRCK DRIVE
	95.0000						
8 BOX	2009.9962	100.0000	559.7262	-4219.9925	0.0000	115.5474	TRACK TOP
	0.5474	0.0000	19.9925	0.0000	-100.0000	0.0000	
9 RCC	2000.0000	0.0000	500.0000	0.0000	100.0000	0.0000	IDLER DUM
	60.0000						
10 RCC	-2200.0000	0.0000	600.0000	0.0000	100.0000	0.0000	DRIVE DUM
	75.0000						

REGION TABLE

1	1	-4	0	FRNT SLOPE
2	2	-5	0	REAR SLOPE
3	3	-1	-2	TRACK BOT
4	6	4	-9	TRCK IDLER
5	7	5	-10	TRCK DRIVE
6	8	-4	-5	TRACK TOP

15. Program Name: SOLIDROT

Description: This program rotates the COM-GEOM solids (except ARS) about any point in the XY, XZ, or YZ planes. Positive rotation is from positive axis to positive axis - that is, positive rotation in the XZ plane is from the positive X axis towards the positive Z axis. User must beware of the direction of the rotation.



Memory: 7539 bytes

Restrictions: The COM-GEOM solid ARS (triangular surfaced polyhedron) cannot be rotated by this program. The large amount of input required was the reason the ARS was not included. If needed however, it would not be that difficult to add.

All rectangular parallelpipeds (RPP) are changed to boxes (BOX) whether or not the rotation is such that they remain RPPs.

Instructions: After the program is loaded, enter the following:

(a) Plane of rotation where

1 = XY plane
2 = XZ plane
3 = YZ plane CR/LF

(b) Angle of rotation (in degrees) CR/LF

(c) X,Y,Z coordinates of point about which you wish to rotate the solid CR/LF

(d) Next input the solid type you wish to rotate, following the instructions on the scope.

(e) Results will then be printed out

(f) To run another case, key CONTINUE CR/LF and program will return to step (a).

Sample Outputs: **The first four sample runs indicate the direction of positive rotation for the three planes of rotation available.**

```
ANGLE OF ROTATION=  90.0000 DEG IN THE  XY  PLANE
PT AROUND WHICH SOLID WAS ROTATED  X=    0.0000 Y=    0.0000 Z=    0.0000
INPUT SOLID
SPH      10.0000      0.0000      0.0000      1.0000
ROTATED SOLID
SPH       0.0000     10.0000      0.0000      1.0000


ANGLE OF ROTATION=  90.0000 DEG IN THE  XZ  PLANE
PT AROUND WHICH SOLID WAS ROTATED  X=    0.0000 Y=    0.0000 Z=    0.0000
INPUT SOLID
SPH      10.0000      0.0000      0.0000      1.0000
ROTATED SOLID
SPH       0.0000      0.0000     10.0000      1.0000


ANGLE OF ROTATION=  90.0000 DEG IN THE  YZ  PLANE
PT AROUND WHICH SOLID WAS ROTATED  X=    0.0000 Y=    0.0000 Z=    0.0000
INPUT SOLID
SPH       0.0000     10.0000      0.0000      1.0000
ROTATED SOLID
SPH       0.0000      0.0000     10.0000      1.0000


ANGLE OF ROTATION=  90.0000 DEG IN THE  YZ  PLANE
PT AROUND WHICH SOLID WAS ROTATED  X=    0.0000 Y=    0.0000 Z=    0.0000
INPUT SOLID
SPH       0.0000      0.0000     10.0000      1.0000
ROTATED SOLID
SPH       0.0000    -10.0000      0.0000      1.0000


ANGLE OF ROTATION=  31.2500 DEG IN THE  YZ  PLANE
PT AROUND WHICH SOLID WAS ROTATED  X=   25.0000 Y=   36.2140 Z= 1236.2540
INPUT SOLID
RPP     -56.2358     25.3247   -987.2354   -350.2140   2365.0000   3214.2300
ROTATED SOLID
BOX     -56.2358  -1424.3902   1670.2941     81.5605      0.0000      0.0000
          0.0000    544.5971    330.4696      0.0000   -440.5578    726.9160
```

16. Program Name: PLOTSOL

Description: This program produces scaled plots of selected COM-GEOM solids. The following solids can be plotted by this program: RPP, BOX, RAW, ARB8, ARB7, ARB6, ARB5, ARB4, RCC, TRC and SPH. Features of this program are as follows:

(1) Ability to replot the same solids at a different aspect (view).
(2) Ability to add solids as long as the limits on the maximum storage requirements are not violated (see restrictions below).
(3) As each solid is entered, the user acknowledges that the input is correct before entering the next solid. This feature helps eliminate input errors.
(4) The scale and azimuth and elevation angle are printed out on each view plotted.
(5) The program gives the user the opportunity to adjust the lengths of the axes so they will be equal. If the axes' lengths are equal then the scale printed out would be correct and circles will look like circles.

Memory: 32053 bytes

Restrictions: The following are the limitations and restrictions of this program:

(1) The sum of the number of TRCs and RCCs to be plotted must not be greater than 5.
(2) The maximum number of SPHs that can be plotted is 5.
(3) The storage requirements for the RPP, BOX ARB8 is 8; for the RAW and ARB6 is 6; for the ARB5 is 5; and for the ARB4 is 4. Using these storage numbers, the sum of all storage requirements for RPPs, BOXs, ARBs and RAWs must not exceed 120. For example, the maximum number of ARB4s that can be plotted is 30.
(4) This program contains no hidden line routine.
(5) This program takes about 55 seconds to process a TRC and about 35 seconds for a RCC.

Instructions: After the program is loaded, perform the following steps, following the instructions on the screen:

(a) Ready the plotter
(b) Enter the number of solids to plot CR/LF
(c) Enter the azimuth and elevation angle for this view CR/LF

(d) Enter the solid type using the code displayed on the screen (1 = RPP, 2 = BOX, etc.) CR/LF

(e) Enter the parameters for the solid as they are asked for

(f) When finished entering the solid parameters, enter 1 if data is correct, 0 if not. If data is correct program processes the data. If data is incorrect program returns to step (e).

(g) Enter the remaining solids using the scheme above (steps (d), (e), and (f)).

(h) When solids are all entered, the program asks if the axes lengths are adjusted. The first time through the program, the user should indicate NO (by entering zero) and two perpendicular lines will be drawn by the plotter. If these lines are of equal length then the axes are of equal length. If these lines are not equal, then adjust them using the scale adjust controls on the plotter. The program will again ask if axes lengths are adjusted and the user continues to say NO and adjust until they are equal.

(i) When axes are adjusted the user should change paper on the plotter and key CONTINUE CR/LF to plot.

(j) Once the solids are plotted, indicate if another view of the same solids is desired, if yes program returns to step (i).

(k) If another view is not desired, the user must next indicate if he wishes to add more solids. If yes, indicate the number to add and the program returns to step (d). If not, the program returns to step (a).

Sample Outputs: Figure 1 shows individual plots of the 11 COM-GEOM solids which this program plots. These plots are intended to familiarize the user with the way these solids look when plotted. Figure 2 is a sample plot of 5 solids at 90° azimuth and 0° elevation. Figure 3 is a plot of the same solids at a different aspect (125° azimuth and 18° elevation).

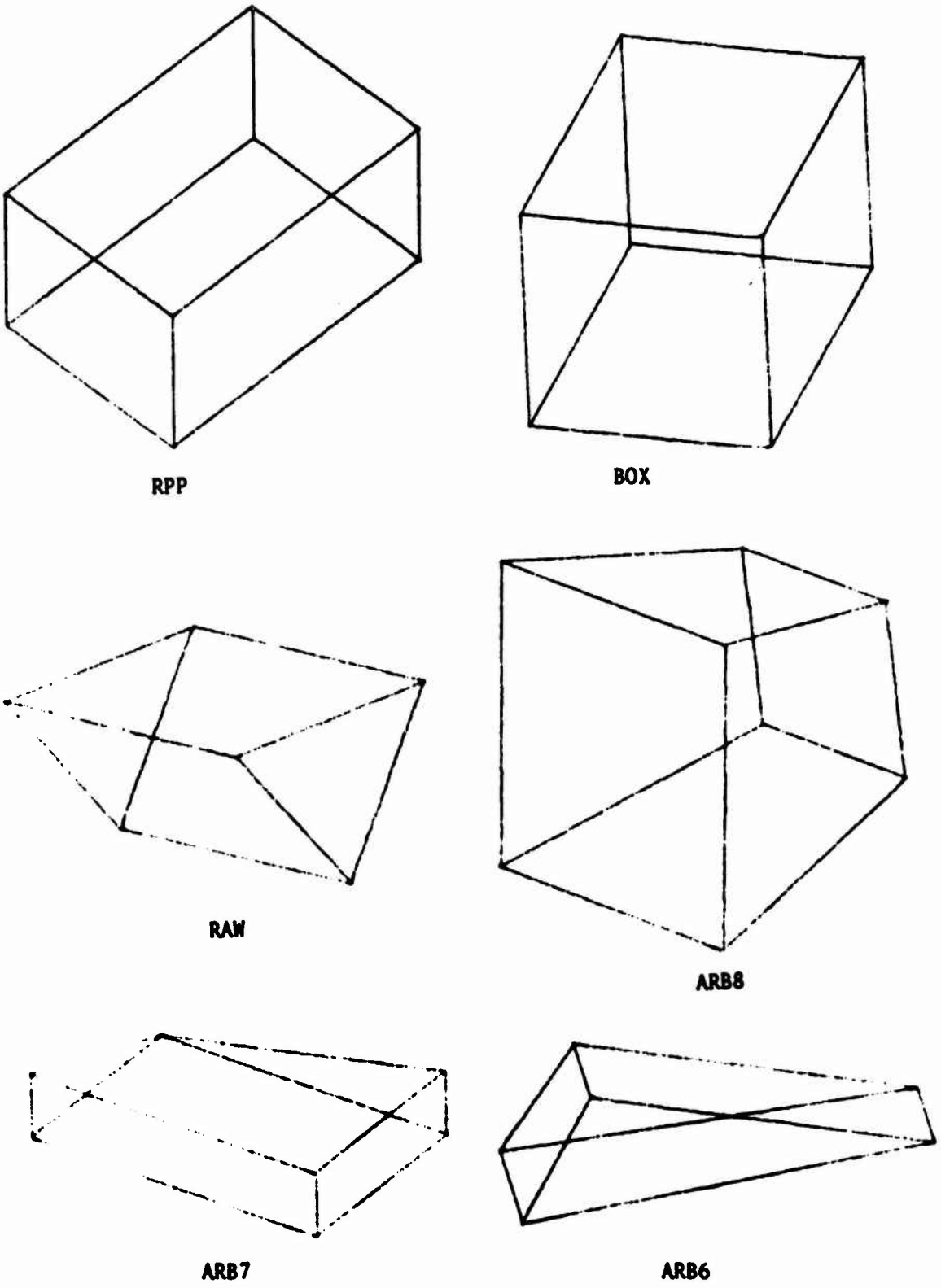
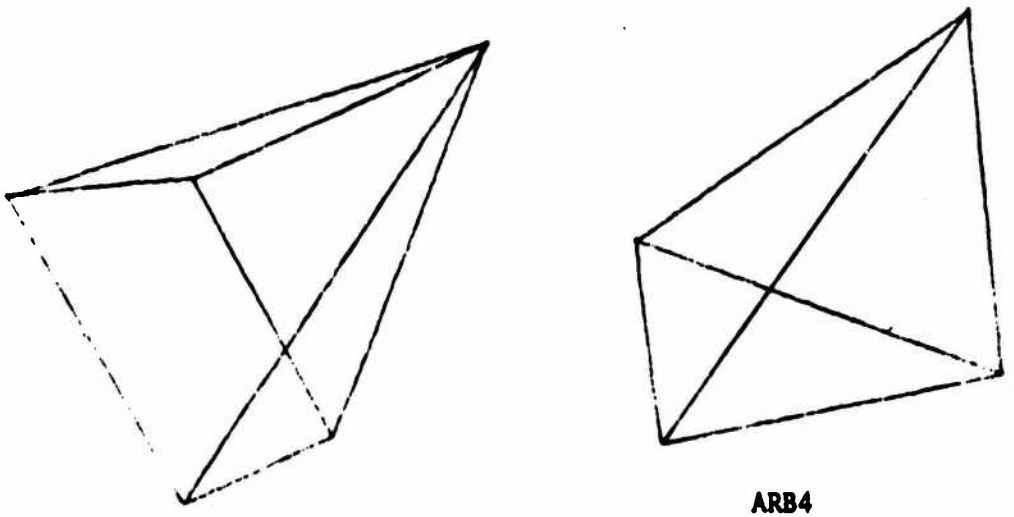
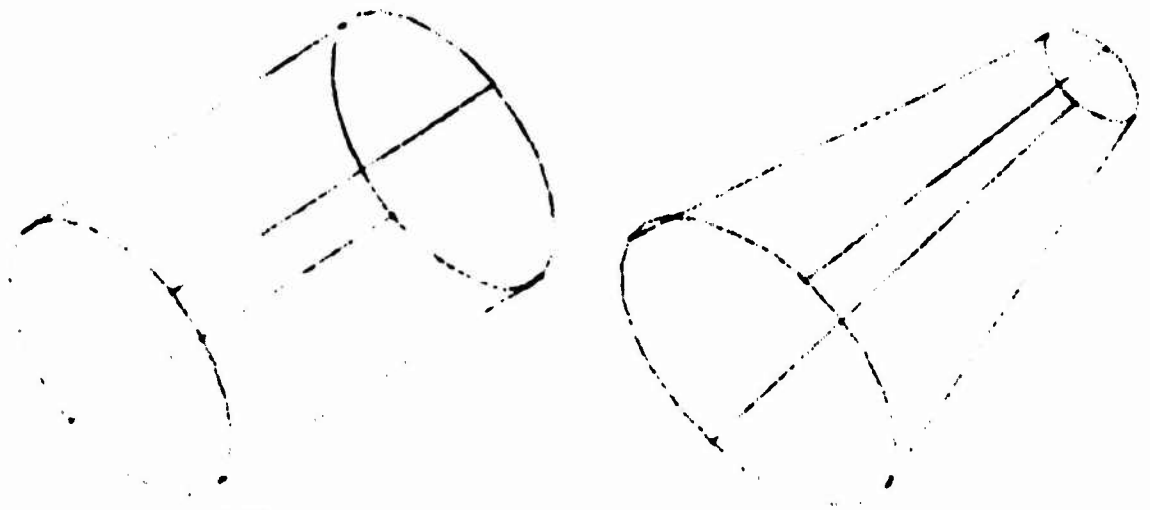


Figure 1. Sample Plots of the Available COM-GEOM Solids



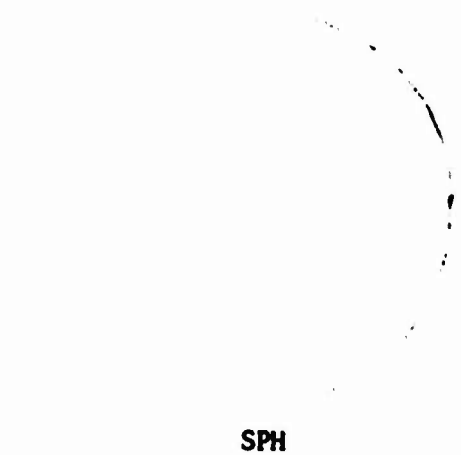
ARB5

ARB4



RCC

TRC



SPH

Figure 1. Sample Plots of the Available COM-GEOM Solids (Continued)

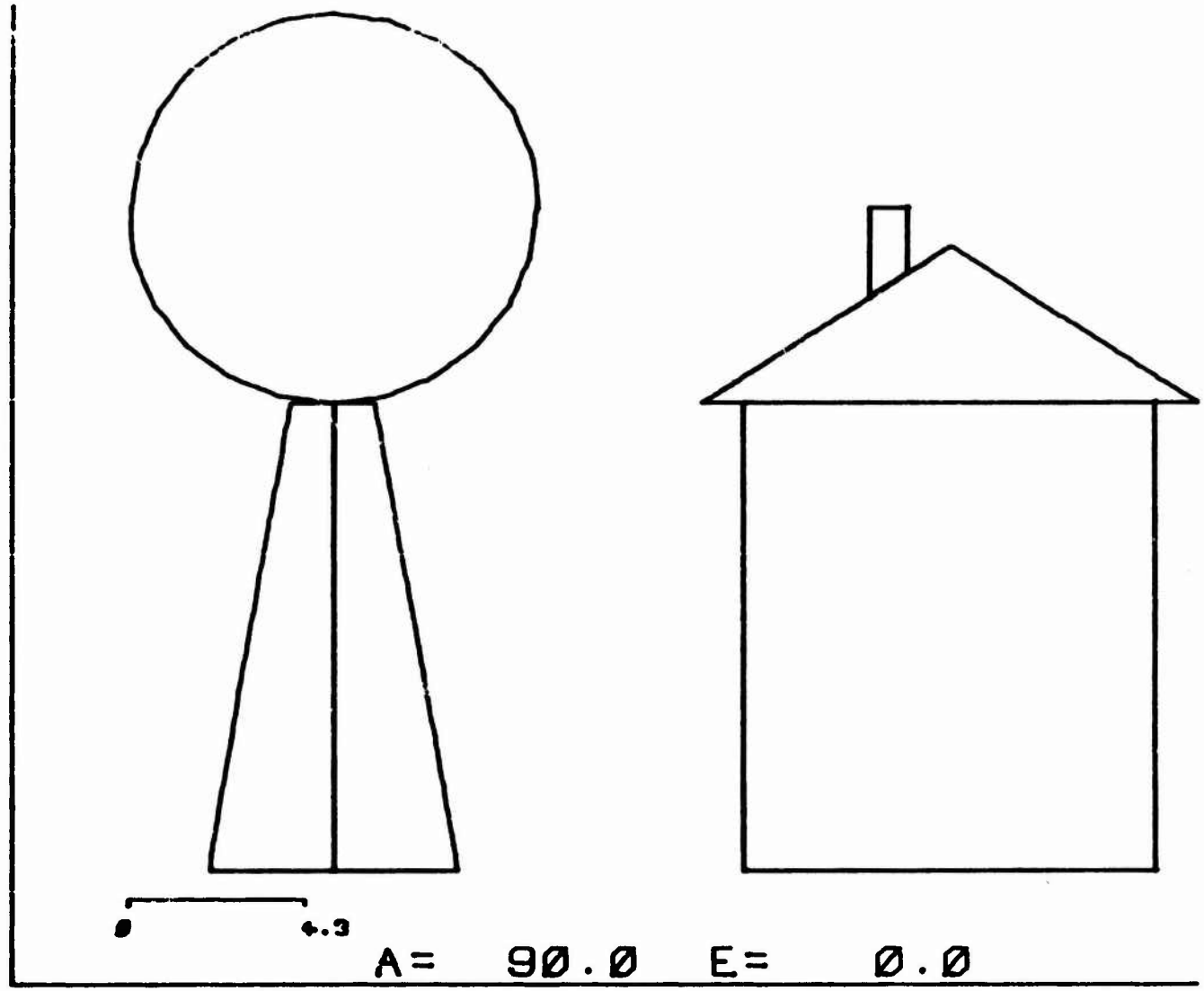


Figure 2. Sample Output of the PLOTSOL Program

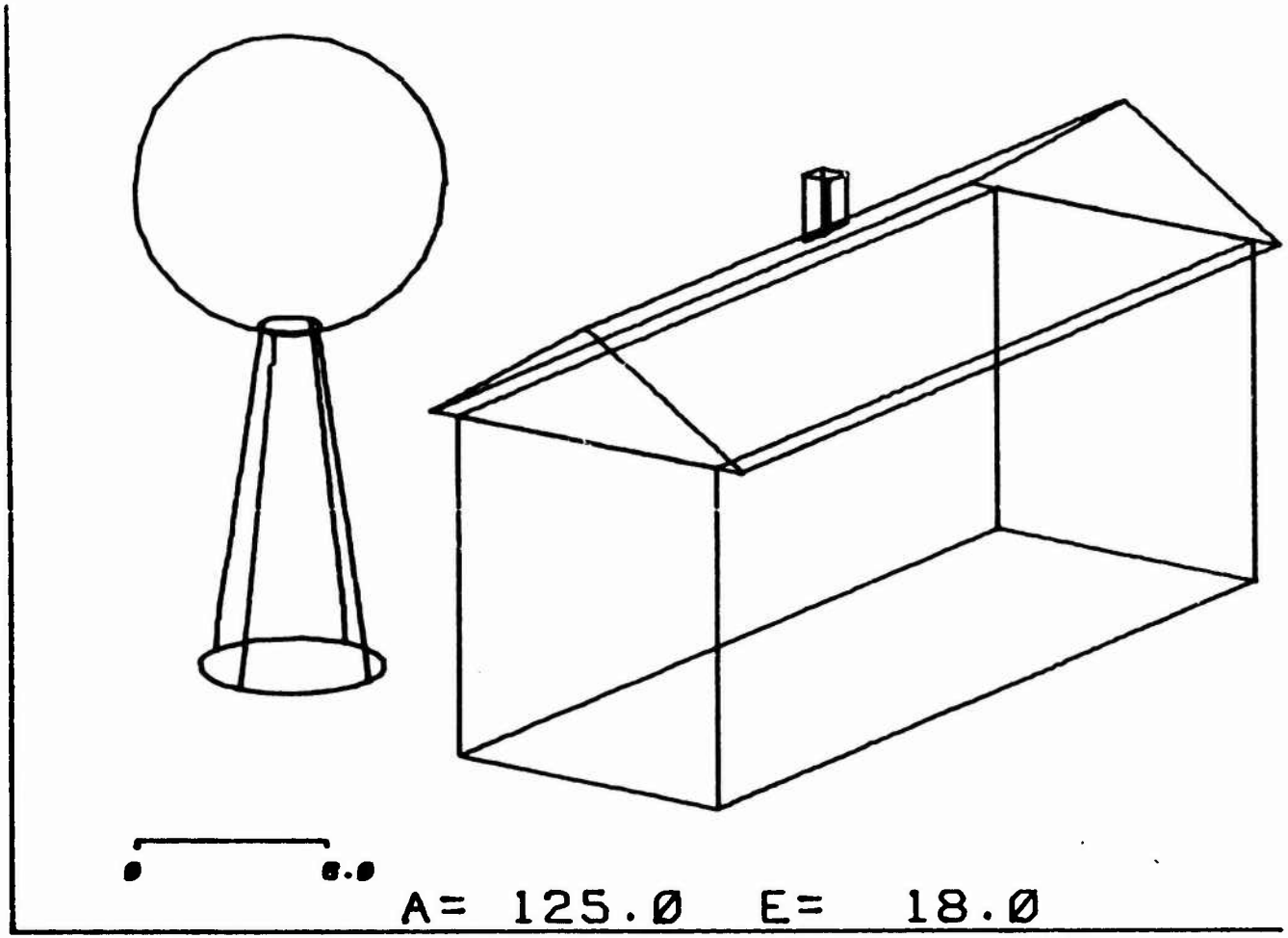


Figure 3. Sample Output of the PLOTSOL Program

17. **Program Name:** DARBIN

Description: This program corrects an Arbitrary Convex Polyhedron (ARB) which has a fourth vertice not in a plane defined by the other three vertices and computes an inside ARB given a thickness for each face.

Memory: 3432 bytes

Restrictions: When solving for the error denoted by the message "FOUR POINTS NOT CO-PLANAR IN FACE 1234", the program will change the X,Y and Z of point 4 to correct the ARB. To change the X,Y and Z of point 1, enter the face ordinal numbers 2341 instead of 1234.

Instructions: After the program is loaded, enter the following:

(a) The number of vertices CR/LF
(b) The number of faces CR/LF
(c) The X,Y, and Z coordinates of each vertice CR/LF
(d) A four digit ordinal number for each face CR/LF. (Face 512 would be entered as 5120) Figure 4 contains face ordinal numbers which may be used.
(e) The option indicator

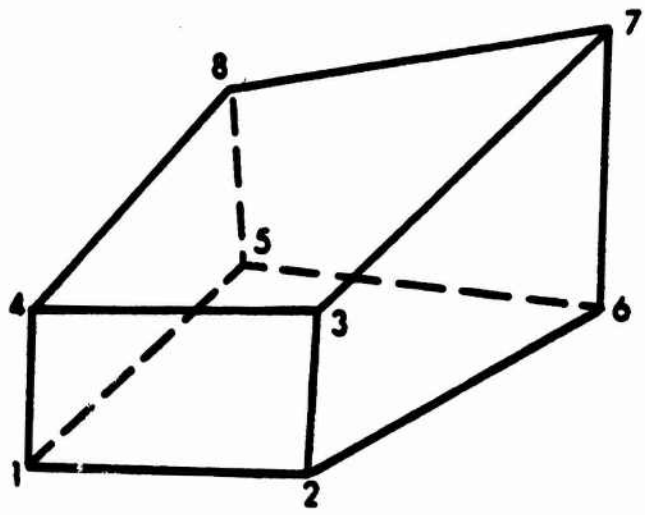
Indicator	Option
1	compute inside ARB
0	return to step (a) CR/LF

(f) Thickness for each face CR/LF
(g) The program returns to step (a).

Sample Outputs:

```
INPUT
     0.0000     0.0000     0.0000     0.0000     5.0000     0.0000
     5.0000     5.0000     0.0000     5.0000     0.0000     0.0000
     0.0000     0.0000     5.0000     0.0000     5.0000     5.0000
     5.0000     5.0000     5.0000     5.0000     0.0000     5.0000
 1234 5678 2376 1485 1265 4378
THICKNESS ARE
     0.1000     0.2000     0.3000    -0.4000     0.5000     0.6000
INSIDE ARB
     4.5000    -0.4000     0.1000     0.5000     4.7000     0.1000
     4.4000     4.7000     0.1000     4.4000    -0.4000     0.1000
     4.5000    -0.4000     4.3000     0.5000     4.7000     4.3000
     4.4000     4.7000     4.3000     4.4000    -0.4000     4.3000
```

```
INPUT
     1.0000      0.0000      0.0000      0.0000      5.0000      0.0000
     5.0000      5.0000      0.0000      5.0000      0.0000      0.0000
     0.0000      0.0000      5.0000      0.0000      5.0000      5.0000
     5.0000      5.0000      5.0000      5.0000      0.0000      5.0000
 1234 5678 2376 1485 1265 4378
FOUR POINTS NOT CO-PLANAR IN FACE 1 2 6 5  DH=    -0.0005
SOLUTION
     1.0000      0.0000      0.0000      0.0000      5.0000      0.0000
     5.0000      5.0000      0.0000      5.0000      0.0000      0.0000
     1.0000      0.0000      5.0000      0.0000      5.0000      5.0000
     5.0000      5.0000      5.0000      5.0000      0.0000      5.0000

INPUT
     1.0000      0.0000      0.0000      0.0000      5.0000      0.0000
     5.0000      5.0000      0.0000      5.0000      0.0000      0.0000
     0.0000      0.0000      5.0000      0.0000      5.0000      5.0000
     5.0000      5.0000      5.0000      5.0000      0.0000      5.0000
 2341 5673 2376 4561 3651 4378
FOUR POINTS NOT CO-PLANAR IN FACE 2 6 5 1  DH=     0.0005
SOLUTION
     0.0000      0.0000      0.0000      0.0000      5.0000      0.0000
     5.0000      5.0000      0.0000      5.0000      0.0000      0.0000
     0.0000      0.0000      5.0000      0.0000      5.0000      5.0000
     5.0000      5.0000      5.0000      5.0000      0.0000      5.0000
```



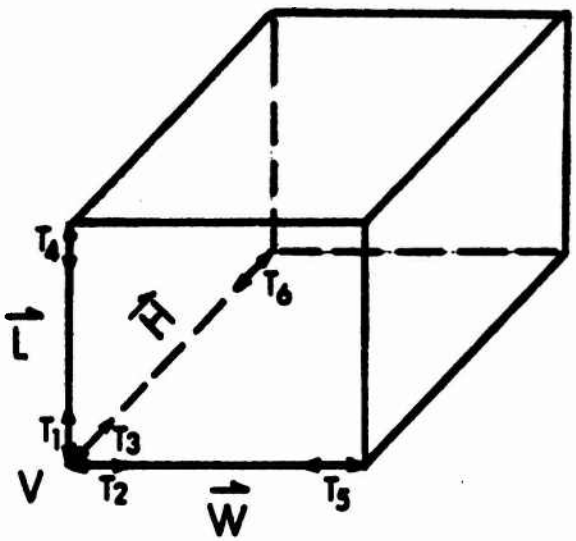
FACE ORDINAL NUMBERS WHICH CAN BE ENTERED

NO. OF VERTICES	NO. OF FACES	T_1	T_2	T_3	T_4	T_5	T_6
8	6	1234	5678	1584	2376	1265	4378
7	6	1234	5670	1450	2376	1265	4375
6	5	1234	2365	1564	5120	6340	
5	5	1234	5120	5230	5340	5410	
4	4	1230	4120	4230	4310		

Figure 4. An Arbitrary Convex Polyhedron (ARB)

18. Program Name: BOXIN

Description: This program corrects a Rectangular Parallelpiped in any orientation (BOX) whose vectors are not normal and computes an inside BOX given a thickness for each face.



Memory: 2717 bytes

Restrictions: A corrected non-normal vector maintains its original length.

Instructions: After the program is loaded, enter the following:

(a) The X,Y and Z coordinates of the vertice CR/LF

(b) The X,Y and Z components of the length ($\overline{L}$) vector CR/LF

(c) The X,Y and Z components of the width ($\overline{W}$) vector CR/LF

(d) The X,Y and Z components of the height ($\overline{H}$) vector CR/LF

(e) The option indicator

Indicator	Option
1	compute inside BOX
0	return to step (a) CR/LF

(f) The thicknesses for each face. CR/LF

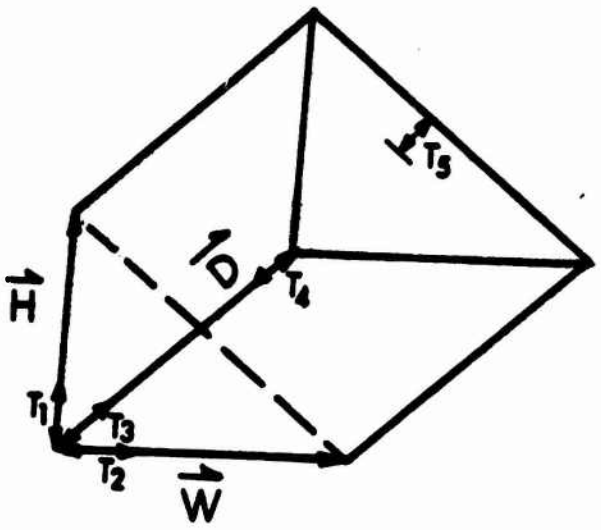
(g) The program returns to step (a).

Sample Outputs:

```
INPUT
     0.0000    0.0000    0.0000   10.0000    0.0000    0.0000
     0.0000   10.0000    0.0000    0.0000    0.0000   10.0000
THICKNESS ARE
     0.1000   -0.2000    0.3000   -0.4000    0.5000   -0.6000
INSIDE BOX
     0.1000   -0.2000    0.3000   10.3000    0.0000    0.0000
     0.0000    9.7000    0.0000    0.0000    0.0000   10.3000
```

```
INPUT
    0.0000    0.0000    0.0000   10.0000    0.0000    0.0000
    0.0000   10.0000    0.0000    1.0000    0.0000   10.0000
HEIGHT AND LENGTH VECTORS ARE NOT NORMAL, ANGLE =  84.28
ARB SOLUTION
    0.0000    0.0000    0.0000   10.0000    0.0000    0.0000
   10.0000   10.0000    0.0000    0.0000   10.0000    0.0000
    1.0000    0.0000   10.0000   11.0000    0.0000   10.0000
   11.0000   10.0000   10.0000    1.0000   10.0000   10.0000
BOX SOLUTION
    0.0000    0.0000    0.0000   10.0000    0.0000    0.0000
    0.0000   10.0000    0.0000    0.0000    0.0000   10.0498
THICKNESS ARE
    0.1000    0.1000    0.2000    0.2000    0.3000    0.3000
INSIDE BOX
    0.1000    0.1000    0.2000    9.7000    0.0000    0.0000
    0.0000    9.6000    0.0000    0.0000    0.0000    9.5498
```

19. Program Name: RAWIN
Description: This program corrects a Right Angle Wedge (RAW) whose vectors are not normal and computes an inside RAW given a thickness for each face.



Memory: 2872 bytes
Restrictions: A corrected non-normal vector maintains its original length.
Instructions: After the program is loaded, enter the following:

(a) The X,Y and Z coordinates of the vertice CR/LF
(b) The X,Y and Z components of the height ($\vec{H}$) vector CR/LF
(c) The X,Y and Z components of the width ($\vec{W}$) vector CR/LF
(d) The X,Y and Z components of the depth ($\vec{D}$) vector CR/LF
(e) The option indicator

Indicator	Option
1	compute inside RAW
0	return to step (a) CR/LF

(f) The thickness for each face CR/LF
(g) The program returns to step (a).

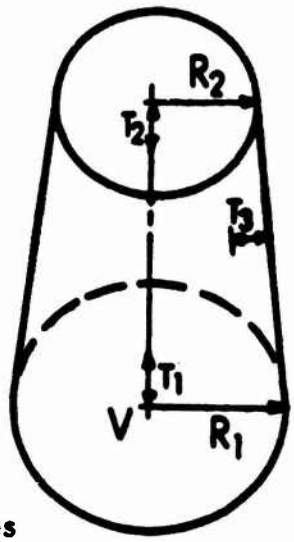
Sample Outputs:

```
INPUT
    0.0000    0.0000    0.0000    0.0000    0.0000   10.0000
    0.0000   10.0000    0.0000   10.0000    0.0000    0.0000
THICKNESS ARE
    0.2000    0.4000    0.6000    0.8000   -1.0000
INSIDE RAW
    0.6000    0.4000    0.2000    0.0000    0.0000   11.2142
    0.0000   11.0142    0.0000    0.6000    0.0000    0.0000
```

```
INPUT
    0.0000     0.0000     0.0000     1.0000     0.0000    10.0000
    0.0000    10.0000     0.0000    10.0000     0.0000     0.0000
DEPTH  AND HEIGHT VECTORS ARE NOT NORMAL, ANGLE =  84.28
ARB SOLUTION
    0.0000     0.0000     0.0000     1.0000     0.0000    10.0000
   11.0000     0.0000    10.0000    10.0000     0.0000     0.0000
    0.0000    10.0000     0.0000    10.0000    10.0000     0.0000
RAW SOLUTION
    0.0000     0.0000     0.0000     1.0000     0.0000    10.0000
    0.0000    10.0000     0.0000     9.9503     0.0000    -0.9950
THICKNESS ARE
    0.5000     0.4000     0.3000     0.2000     0.1000
INSIDE RAW
    0.3482     0.4000     0.4676     0.9361     0.0000     9.3614
    0.0000     9.4589     0.0000     9.4528     0.0000    -0.9452
```

20. Program Name: TRCIN

Description: This program computes an inside Truncated Right Angled Cone (TRC) given a thickness for each surface.



Memory: 1047 bytes

Restrictions: None

Instructions: After the program is loaded, enter the following:

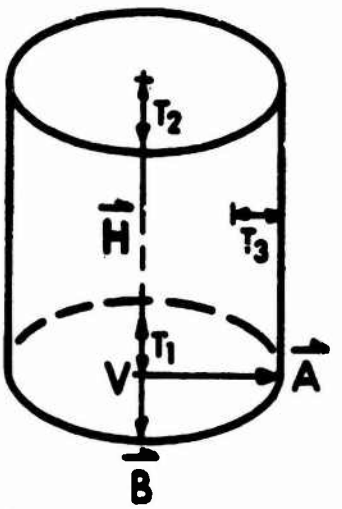
(a) The X,Y and Z coordinates of the vertice CR/LF
(b) The X,Y and Z components of the height vector CR/LF
(c) The radius of the base CR/LF
(d) The radius of the top CR/LF
(e) The thickness for each surface CR/LF
(f) The program returns to step (a).

Sample Outputs:

```
INPUT
     0.0000      0.0000      0.0000      0.0000      0.0000     10.0000
    10.0000      5.0000
THICKNESSES ARE
    -1.0000      2.0000      3.0000
INSIDE TRC
     0.0000      0.0000     -1.0000      0.0000      0.0000      9.0000
     7.1458      2.6458

INPUT
     0.0000      0.0000      0.0000      0.0000      0.0000     10.0000
    10.0000      1.0000
THICKNESSES ARE
     1.0000      2.0000      3.0000
WARNING, TOP RADIUS OF INSIDE TRC =    -1.23
RADIUS RESET TO .0001 AND THICKNESS(2) = 0
INSIDE TRC
     0.0000      0.0000      1.0000      0.0000      0.0000      5.6205
     5.0639      0.0001
```

21. **Program Name:** RECIN

Description: This program corrects a Right Angle Elliptical Cylinder (REC) whose vectors are not normal and computes an inside REC given a thickness for each surface.



Memory: 2337 bytes

Restrictions: The corrected non-normal vector maintains its original length. The inside elliptical surface does not yield a constant thickness. This is especially true when the ratio of the lengths of the minor axis to the major axis is less than 0.8. Therefore, this program computes an inside REC with an elliptical surface which yields an average thickness equal to the thickness entered.

Instructions: After the program is loaded, enter the following:

(a) The X,Y and Z coordinates of the vertice CR/LF

(b) The X,Y and Z components of the height vector CR/LF

(c) The X,Y and Z components of the defining the semi-major axis CR/LF

(d) The X,Y and Z components of the vector defining the semi-minor axis CR/LF

(e) The option indicator

Indicator	Option
1	compute inside REC
0	return to step (a) CR/LF

(f) The thickness for each surface CR/LF

(g) The program returns to step (a).

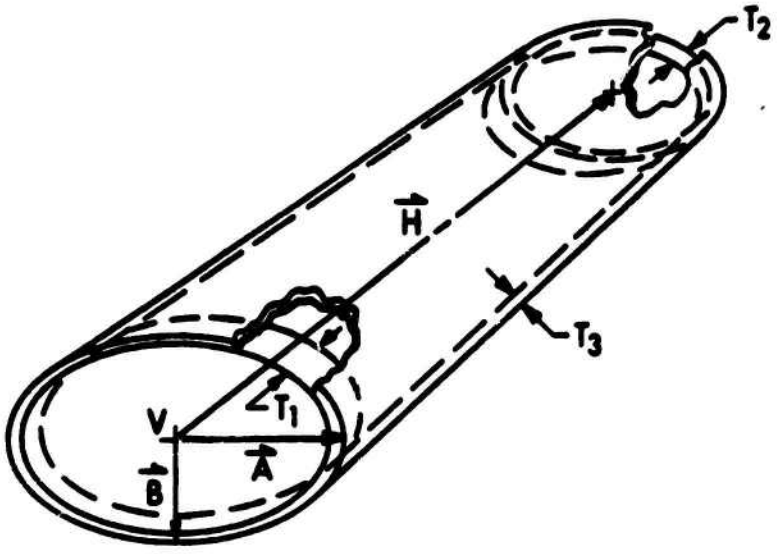
Sample Outputs:

```
INPUT
   0.0000    0.0000    0.0000    0.0000    0.0000   10.0000
  10.0000    0.0000    0.0000    0.0000    5.0000    0.0000
THICKNESS ARE
```

```
    1.0000      2.0000     -3.0000
INSIDE REC
    0.0000      0.0000      1.0000      0.0000      0.0000      7.0000
   13.0000      0.0000      0.0000      0.0000      8.1021      0.0000

INPUT
    0.0000      0.0000      0.0000      0.0000      0.0000     10.0000
   10.0000      1.0000      0.0000      0.0000      5.0000      0.0000
SEMI-MAJOR AND SEMI-MINOR VECTORS NOT NORMAL, ANGLE=  84.28
SOLUTION
    0.0000      0.0000      0.0000      0.0000      0.0000     10.0000
   10.0498      0.0000      0.0000      0.0000      5.0000      0.0000
THICKNESS ARE
    1.0000      2.0000     -3.0000
INSIDE REC
    0.0000      0.0000      1.0000      0.0000      0.0000      7.0000
   13.0498      0.0000      0.0000      0.0000      8.1032      0.0000
```

22. Program Name: TECIN

Description: This program corrects a Truncated Elliptical Cone (TEC) whose base vectors are not normal and computes an inside TEC given a thickness for each surface.



Memory: 3432 bytes

Restrictions: The corrected non-normal vector maintains its original length. The elliptical surface does not yield a constant thickness. This is especially true when the ratio of the lengths of the minor axis to major axis is less than 0.8. Therefore, this program computes an inside TEC with an elliptical surface which yields an average thickness equal to the thickness entered. As a result of the restriction of defining the top of a TEC as a ratio of the base, the inside TEC is a best fit. Therefore, a TEC with a large ratio or a relatively large thickness relative to the length of the minor axis or a small minor axis relative to the major axis may have a large variation of thicknesses on its elliptical surface.

Instructions: After the program is loaded, enter the following:

(a) The X,Y and Z coordinates of the vertice CR/LF

(b) The X,Y and Z components of the height vector CR/LF

(c) The X,Y and Z components of the vector defining the semi-major axis of the base CR/LF

(d) The X,Y and Z components of the vector defining the semi-minor axis of the base CR/LF
(e) The ratio of base to the top CR/LF
(f) The option indicator

Indicator	Option
1	compute inside TEC
0	return to step (a) CR/LF

(g) The thicknesses for base, top and elliptical side in exact order CR/LF
(h) The program returns to step (a).

Sample Outputs:

```
INPUT
     0.0000     0.0000     0.0000     0.0000     5.0000    10.0000
    10.0000     0.0000     0.0000     0.0000     5.0000     0.0000
     2.0000
THICKNESS ARE
    -0.5000     1.0000     2.0000
INSIDE TEC
     0.0000     0.0894    -0.5000     0.0000     4.7500     9.5000
     0.0139     0.0000     0.0000     0.0000     2.6538     0.0000
     4.6607
RATIO OF MAJOR AXIS    2.4552  RATIO OF MINOR AXIS    5.0181
TOP MAJOR AXIS         3.2639     TOP MINOR AXIS         0.5208

INPUT
     0.0000     0.0000     0.0000     0.0000     5.0000    10.0000
    10.0000     5.0000     0.0000     0.0000     5.0000     0.0000
     1.1250
SEMI-MAJOR AND SEMI-MINOR VECTORS NOT NORMAL, ANGLE = 63.43
SOLUTION
     0.0000     0.0000     0.0000     0.0000     5.0000    10.0000
    11.1303     0.0000     0.0000     0.0000     5.0000     0.0000
     1.1250
THICKNESS ARE.
     0.1000     0.1000     0.6000
INSIDE TEC
     0.0000     0.2647     0.1000     0.0000     4.0000     9.0000
    10.5633     1.0000     0.0000     0.0000     4.1331     0.0000
     1.1131
RATIO OF MAJOR AXIS    1.1302  RATIO OF MINOR AXIS    1.1133
TOP MAJOR AXIS         9.2458     TOP MINOR AXIS         3.7122
```

23. Program Name: PARB

Description: This program computes the points of intersection of a set of planes taken three at a time. Each plane can be defined by either three points; azimuth angle, elevation angle and point; or an equation.

Memory: 4026 bytes

Restrictions: This program is limited to a maximum of 25 planes. Combinations of three planes containing parallel planes or whose point of intersection distance from the origin is greater than 100,000 is not computed or printed. The plane equation (AX + BY + CZ = D) is normalized so that a parallel plane can be obtained by only changing the value of D. Furthermore, the difference between the value of D in the original equation and the final equation is the distance between the planes.

Instructions: After the program is loaded, enter the following:

(a) The number of planes CR/LF

(b) The type of input indicator for each plane

Indicator	Input
0	azimuth, elevation and point
1	three points
2	plane equation CR/LF

(c) If the indicator keyed in step (b) is 0, enter the following:

1. The azimuth angle (about z axis) CR/LF
2. The elevation angle (about y axis) CR/LF

(d) If the indicator keyed in step (b) is 1, enter the following:
The X,Y and Z coordinates for each of three points in the plane CR/LF

(e) If the indicator keyed in step (b) is 2, enter the following:
The A,B,C,D coefficients of the plane equation AX + BY + CY = D CR/LF

(f) The program returns to step (b) until all planes are entered.

(g) The option indicator

Indicator	Option
0	change planes previous entered
1	return to step (a) CR/LF

(h) The number of the plane to be corrected CR/LF
If zero is keyed, the program computes the intersections of the planes and returns to step (g). If a number greater than zero is keyed, the program returns to step (b).

Sample Outputs:

```
INPUT FOR FACE  1
AZIMUTH =   30.00  ELEVATION =   26.00
POINT =   120.0000   245.0000    64.0000
PLANE EQ. = 0.77837 X+ 0.44939 Y+ 0.43837 Z =          231.56

INPUT FOR FACE  2
3-POINTS
   -10.0000     0.0000     0.0000   -10.0000     2.0000     5.0000
   -10.0000    30.0000    58.0000
PLANE EQ. =-1.00000 X+ 0.00000 Y+ 0.00000 Z =           10.00

INPUT FOR FACE  3
PLANE EQ. = 0.00000 X+ 0.55470 Y+ 0.83205 Z =          120.47

INPUT FOR FACE  4
AZIMUTH =  210.00  ELEVATION =  -29.00
POINT =   -10.0000    30.0000    58.0000
PLANE EQ. = 0.01511 X+ 0.00872 Y+ 0.99984 Z =           58.10
  PLANES                POSSIBLE SOLUTION
  1  2  3        -10.0000   1099.0472   -580.6981
  1  2  4        -10.0000    479.8491     54.0739
  1  3  4        132.2346    146.8891     54.0739
  2  3  4        -10.0000    142.4724     57.0184

INPUT FOR FACE  4
PLANE EQ. = 0.01511 X+ 0.00872 Y+ 0.99984 Z =           57.10
  PLANES                POSSIBLE SOLUTION
  1  2  3        -10.0000   1099.0472   -580.6981
  1  2  4        -10.0000    480.3316     53.0667
  1  3  4        181.9292    148.4013     53.0655
  2  3  4        -10.0000    143.9933     56.0044
```

APPENDIX A

Geometry Aid Program Listings

The character set of the typewriter used to produce these program listings is such that the following corrections must be made by the user:

1. ! must be replaced by ↑
2. [must be replaced by <
3.] must be replaced by >

Table A-I. SPHERE Program Listing

```
10 DIM P(4,3),C(3,3),D(3)
20 SELECT PRINT 005(64):PRINT "SPHERE PROGRAM"
30 PRINT "FINDS SPHERE DEFINED BY 4 NONPLANAR POINTS":PRINT "OR SURFAC
E OF THE SPHERE"
40 FOR I=1TO 4
50 SELECT PRINT 005(64):PRINT "POINT",I:INPUT "X,Y,Z=",P(I,1),P(I,2),P
(I,3):NEXT I
60 SELECT PRINT 215(80):PRINT "INPUT POINTS"
70 PRINT "PT          X                Y                Z"
80 FOR I=1TO 4:PRINTUSING 90,I,P(I,1),P(I,2),P(I,3):NEXT I
90 # -######.####  -######.####  -######.####
100 FOR I=1TO 3:D(I)=0
110 FOR J=1TO 3:C(I,J)=2*(P(4,J)-P(I,J))
120 D(I)=P(I)+P(4,J)!2-P(I,J)!2:NEXT J:NEXT I
130 FOR K=1TO 2:IF C(K,K)=0THEN 160
140 FOR I=K+1TO 3:F=C(I+1,K)/C(K,K):D(I+1)=D(I+1)-F*D(I)
150 FOR J=K+1TO 3:C(I+1,J)=C(I+1,J)-F*C(K,J):NEXT J:NEXT I:GOTO 190
160 FOR I=K+1TO 3:FOR J=K+1TO 3:C(K,J)=C(K,J)+C(I+1,J):NEXT J
170 D(K)=D(K)+D(I+1):IF C(K,K)[]0THEN 140:NEXT I
180 GOTO 310
190 NEXT K
200 IF C(3,3)=0THEN 310:D(3)=D(3)/C(3,3)
210 D(2)=(D(2)-D(3)*C(2,3))/C(2,2)
220 D(1)=(D(1)-D(2)*C(1,2)-D(3)*C(1,3))/C(1,1)
230 R=0:FOR J=1TO 3:R=R+(P(1,J)-D(J))!2:NEXT J
240 R=SQR(R)
250 SELECT PRINT 215(80):PRINTUSING 270,D(1),D(2),D(3)
260 PRINTUSING 280,R
270 #SPHERE      XC=-######.####   YC=-######.####   ZC=-######.####
280              RADIUS=-######.####
290 PRINT :PRINT :SELECT PRINT 005(64):PRINT "TO RUN AGAIN, KEY CONTIN
UE  EXECUTE"
300 PRINT "TO STOP, KEY CLEAR  EXECUTE":STOP :GOTO 40
310 PRINT "POINTS ARE COPLANAR":GOTO 290
```

Table A-II. CIRCIR Program Listing

```
10 SELECT PRINT 005(64):PRINT "CIRCIR PROGRAM":PRINT "FINDS INTERSECTION
 OF 2 CIRCLES"
20 INPUT "X,Y OF CENTER AND RAD OF CIRCLE 1=",A1,B1,R1
30 IF R1]0THEN 50:PRINT "RADIUS[=0, TRY AGAIN"
40 INPUT "RADIUS=",R1:GOTO 30
50 INPUT "X,Y OF CENTER AND RAD OF CIRCLE 2=",A2,B2,R2
60 IF R2]0THEN 80:PRINT "RADIUS[=0, TRY AGAIN"
70 INPUT "RADIUS=",R2:GOTO 60
80 SELECT PRINT 215(84):PRINTUSING 90,A1,B1,R1:PRINTUSING 100,A2,B2,R2
90%CIRCLE 1    X CENT=-#####.####   Y CENT=-#####.####   R=-#####.####
100%CIRCLE 2          -#####.####          -#####.####     -#####.####
110 IF B1=B2THEN 230
120 M=-(A2-A1)/(B2-B1):B3=(R1!2-R2!2-A1!2+A2!2-B1!2+B2!2)/(2*(B2-B1))
130 A4=1+M!2:B4=2*M*(B3-B1)-2*A1:C4=A1!2+(B3-B1)!2-R1!2
140 D=B4!2-(4*A4*C4):IF D[0THEN 220
150 X4=(-B4+SQR(D))/(2*A4)
160 X5=(-B4-SQR(D))/(2*A4)
170 Y4=M*X4+B3:Y5=M*X5+B3
180 PRINTUSING 190,X4,Y4:PRINTUSING 200,X5,Y5
190%INTERSECTION POINTS    X=-#####.####    Y=-#####.####
200%                         -#####.####      -#####.####
210 PRINT :PRINT :GOTO 20
220 PRINT "NO INTERSECTION":PRINT :PRINT :GOTO 20
230 IF A1[]A2THEN 240:IF R1=R2THEN 290:GOTO 220
240 IF ABS(A2-A1)](R1+R2)THEN 220
250 IF ABS(A2-A1)=(R1+R2)THEN 300
260 X4=(R1!2-R2!2-A1!2+A2!2-B1!2+B2!2)/(2*(A2-A1)):X5=X4
270 Y4=B1+SQR(R1!2-(X4-A1)!2)
280 Y5=B1-SQR(R1!2-(X4-A1)!2):GOTO 180
290 PRINT "SAME CIRCLE":PRINT :PRINT :GOTO 20
300 X4=A1+R1:Y4=B1
310 PRINTUSING 320,X4,Y4:PRINT :PRINT :GOTO 20
320%CIRCLES TANGENT AT     X=-#####.####    Y=-#####.####
330 STOP :END
```

Table A-III. CIRCLE Program Listing

```
10 PRINT "CIRCLE PROGRAM":PRINT "FINDS CIRCLE DEFINED BY 3 NONCOLLINEA
R POINTS"
20 DIM X(10),Y(10)
30 FOR J=1TO 3:PRINT "POINT",J:INPUT "X=",X(J)
40 INPUT "Y=",Y(J):NEXT J
50 A=X(1)-X(2):B=Y(1)-Y(2)
60 C=X(2)-X(3):D=Y(2)-Y(3)
70 E=(Y(1)!2+X(1)!2-Y(2)!2-X(2)!2)/2
80 F=(Y(2)!2+X(2)!2-Y(3)!2-X(3)!2)/2
90 D1=A*D-C*B
100 IF D1[]OTHEN 110:PRINT "***POINTS COLLINEAR***":GOTO 30
110 X1=(E*D-F*B)/D1:Y1=(A*F-E*C)/D1
120 R=SQR((X(1)-X1)!2+(Y(1)-Y1)!2)
130 SELECT PRINT 211(85)
140 PRINT "          INPUT POINTS"
150 PRINT "      X                  Y"
160 FOR I=1TO 3:PRINTUSING 170,X(I),Y(I):NEXT I
170%-#####.####     -#####.####
180 PRINTUSING 190,X1,Y1,R
190%XCENT=-#####.####   YCENT=-#####.####   RAD=-#####.####
200 PRINT :PRINT
210 SELECT PRINT 005(64)
220 PRINT "TO RUN AGAIN, KEY CONTINUE  EXECUTE"
230 PRINT "TO STOP, KEY CLEAR  EXECUTE"
240 STOP
250 GOTO 30
```

Table A-IV. RCC Program Listing

```
10 SELECT PRINT 005(64):PRINT "RCC PROGRAM"
20 PRINT "FINDS RCC WITH BASE DEFINED BY 3 POINTS ON CIRCUMFERENCE"
30 DIM A(10,11),V(10),X(5),Y(5),Z(5),B(5)
40 PRINT "INPUT THE 3 POINTS"
50 FOR I=1TO 3:PRINT "POINT",I
60 INPUT "X,Y,Z=",X(I),Y(I),Z(I):NEXT I
70 INPUT "LENGTH OF HEIGHT VECTOR",H:N=3
80 IF H]0THEN 100
90 PRINT "***** E R R O R   H E I G H T  [ =  0 *****":GOTO 70
100 U1=X(2)-X(1):U2=Y(2)-Y(1):U3=Z(2)-Z(1)
110 V1=X(3)-X(2):V2=Y(3)-Y(2):V3=Z(3)-Z(2)
120 W1=X(3)-X(1):W2=Y(3)-Y(1):W3=Z(3)-Z(1)
130 A(1,1)=U1:A(1,2)=U2:A(1,3)=U3
140 A(2,1)=V1:A(2,2)=V2:A(2,3)=V3
150 H1=V2*W3-V3*W2:H2=-(V1*W3-V3*W1):H3=V1*W2-V2*W1
160 H4=SQR(H1!2+H2!2+H3!2):IF H4]0THEN 180
170 PRINT "***** E R R O R  -  P T S   C O L L I N E A R *****":GOTO 40
180 H1=H*H1/H4:H2=H2*H/H4:H3=H*H3/H4
190 A(3,1)=H1:A(3,2)=H2:A(3,3)=H3
200 FOR S=1TO N
210 FOR T=S TO N:IF A(T,S)[]0 THEN 230:NEXT T
220 PRINT "***** E R R O R   P T S   C O L L I N E A R *****":GOTO 40
230 GOSUB 330
240 A(S,S)=1/A(S,S):GOSUB 360
250 FOR T=1 TO N:IF T=S THEN 270
260 B=-A(T,S): A(T,S)=0:GOSUB 390
270 NEXT T:NEXT S
280 FOR S=N TO 1 STEP -1:IF V(S)=S THEN 320
290 FOR J=1 TO N
300 B=A(J,S): A(J,S)=A(J,V(S)): A(J,V(S))=B
310 NEXT J
320 NEXT S:GOTO 420
330 FOR J=1 TO N
340 B=A(S,J): A(S,J)=A(T,J): A(T,J)=B
350 NEXT J: V(S)=T:RETURN
360 FOR J=1 TO N:IF J=S THEN 380
370 A(S,J)=A(S,S)*A(S,J)
380 NEXT J:RETURN
390 FOR J=1 TO N
400 A(T,J)=A(T,J)+B*A(S,J)
410 NEXT J:RETURN
420 B(1)=(X(2)!2-X(1)!2+Y(2)!2-Y(1)!2+Z(2)!2-Z(1)!2)/2
430 B(2)=(X(3)!2-X(2)!2+Y(3)!2-Y(2)!2+Z(3)!2-Z(2)!2)/2
440 B(3)=H1*X(1)+H2*Y(1)+H3*Z(1)
450 X9=A(1,1)*B(1)+A(1,2)*B(2)+A(1,3)*B(3)
460 Y9=A(2,1)*B(1)+A(2,2)*B(2)+A(2,3)*B(3)
470 Z9=A(3,1)*B(1)+A(3,2)*B(2)+A(3,3)*B(3)
```

Table A-IV. RCC Program Listing (Continued)

```
480 R9=SQR((X(1)-X9)!2+(Y(1)-Y9)!2+(Z(1)-Z9)!2)
490 SELECT PRINT 215(90)
500 PRINT "INPUT POINTS:"
510 FOR I=1TO 3:PRINTUSING 520,I,X(I),Y(I),Z(I):NEXT I
520 %POINT #    X=-#####.####   Y=-#####.####   Z=-#####.####
530 PRINT "LENGTH OF HEIGHT VECTOR=",H:PRINT
540 PRINT "THE PARAMETERS OF THE RCC:"
550 PRINTUSING 560,X9,Y9,Z9
560 %CENTER OF BASE  XC=-#####.#### YC=-#####.#### ZC=-#####.####
570 PRINTUSING 580,H1,H2,H3
580 %HEIGHT VECTOR  DX=-#####.#### DY=-#####.#### DZ=-#####.####
590 PRINTUSING 600,-H1,-H2,-H3
600 %      OR       DX=-#####.#### DY=-#####.#### DZ=-#####.####
610 PRINTUSING 620,R9:PRINT :PRINT :PRINT
620 %RADIUS OF BASE=-#####.####
630 SELECT PRINT 005(64): GOTO 40
```

Table A-V. LINECIR Program Listing

```
10 SELECT PRINT 005(64)
20 PRINT "LINE CIRCLE PROGRAM"
30 PRINT "FINDS INTERSECTION OF LINE AND CIRCLE"
40 PRINT "LINE INPUT:"
50 INPUT "X,Y,DEL X,DEL Y=",X1,Y1,D1,D2
60 PRINT "CIRCLE INPUT:"
70 INPUT "X,Y OF CENTER, RADIUS=",X2,Y2,R
80 IF R]0THEN 100:PRINT "ERROR RADIUS [= 0, TRY AGAIN"
90 INPUT "RADIUS=",R:GOTO 80
100 SELECT PRINT 215(80)
110 PRINTUSING 120,X1,Y1
120%LINE INPUT       X=-#####.####       Y=-#####.####
130 PRINTUSING 140,D1,D2
140%              DEL X=-#####.####   DEL Y=-#####.####
150 PRINTUSING 160,X2,Y2,R
160%CIRCLE INPUT   XC=-#####.####  YC=-#####.####  R=-#####.####
170 IF D1=0THEN 300:S=D2/D1:Y3=Y1-S*X1:A=1.+S!2
180 B=2.*S*(Y3-Y2)-2.*X2:C=X2!2+(Y3-Y2)!2-R!2
190 D3=B!2-4.*A*C:IF D3[0.THEN 360:IF D3=0.THEN 370
200 X4=(-B+SQR(D3))/(2.*A):X5=(-B-SQR(D3))/(2.*A)
210 SELECT PRINT 215(80)
220 Y4=S*X4+Y3:Y5=S*X5+Y3:PRINT "INTERSECTION POINTS"
230 PRINTUSING 240,X4,Y4:PRINTUSING 240,X5,Y5
240%X=-#####.####   Y=-#####.####
250 PRINT :PRINT
260 SELECT PRINT 005(64)
270 PRINT "TO STOP, KEY CLEAR CR-LF"
280 PRINT "TO RUN AGAIN, KEY CONTINUE CR-LF"
290 STOP :GOTO 40
300 IF D2=0.THEN 350
310 D4=R!2-(X1-X2)!2:IF D4=0THEN 330:IF D4[0THEN 360
320 X4=X1:X5=X1:Y4=Y2+SQR(D4):Y5=Y2-SQR(D4):GOTO 340
330 X4=X1:Y4=Y2:PRINTUSING 380,X4,Y4:GOTO 260
340 PRINT "INTERSECTION POINTS":GOTO 230
350 PRINT "INPUT ERROR, SLOPE UNDEFINED":PRINT :PRINT :GOTO 260
360 PRINT "NO INTERSECTION":PRINT :PRINT :GOTO 250
370 X4=-B/(2.*A):Y4=S*X4+Y3:PRINTUSING 380,X4,Y4
380%LINE TANGENT TO CIRCLE AT   X=-#####.####   Y=-#####.####
390 GOTO 250
400 STOP
```

Table A-VI. TANCIR Program Listing

```
10 SELECT PRINT 005(64)
20 PRINT "TANCIR PROGRAM"
30 PRINT "FINDS TANGENT PTS ON A CIRCLE FROM A PT OUTSIDE"
40 INPUT "INPUT POINT X,Y",X1,Y1
50 INPUT "INPUT CIRCLE CENTER X,Y AND RADIUS:",X2,Y2,R
60 SELECT PRINT 005(64)
70 IF R]0THEN 80:PRINT "ERROR RADIUS[=0, TRY AGAIN":INPUT "RADIUS=",R:
GOTO 70
80 SELECT PRINT 215(80):PRINTUSING 90,X1,Y1
90%POINT INPUT  X=-#####.####  Y=-#####.####
100 PRINTUSING 110,X2,Y2,R
110%CIRCLE INPUT  XCENT=-#####.####  YCENT=-#####.####  R=-#####.####
120 A=(X1-X2)!2+(Y1-Y2)!2
130 IF SQR(A)[RTHEN 280
140 IF SQR(A)=RTHEN 300
150 X3=X2+(((R!2)*(X1-X2))-(R*(Y1-Y2)*SQR(A-R!2)))/A
160 X4=X2+(((R!2)*(X1-X2))+(R*(Y1-Y2)*SQR(A-R!2)))/A
170 Y3=Y2+(((R!2)*(Y1-Y2))+(R*(X1-X2)*SQR(A-R!2)))/A
180 Y4=Y2+(((R!2)*(Y1-Y2))-(R*(X1-X2)*SQR(A-R!2)))/A
190 PRINTUSING 200,X3,Y3
200%TANGENT PTS   X=-#####.####  Y=-#####.####
210 PRINTUSING 220,X4,Y4:PRINT :PRINT
220%              X=-#####.####  Y=-#####.####
230 SELECT PRINT 005(64)
240 PRINT "TO RUN AGAIN, KEY CONTINUE,CR-LF"
250 PRINT "TO STOP, KEY CLEAR,CR-LF"
260 STOP
270 GOTO 40
280 PRINT "POINT IS INSIDE CIRCLE":PRINT :PRINT
290 GOTO 230
300 PRINT "POINT IS ON CIRCLE":PRINT :PRINT
310 GOTO 230
```

Table A-VII. PLANEINT Program Listing

```
10 DIM X(10),Y(10),Z(10),A(5,5)
20 SELECT D
30 SELECT PRINT 005(64)
40 PRINT "PLANEINT PROGRAM":PRINT "FINDS INTERSECTION POINT OF THREE P
LANES"
50 PRINT "EACH PLANE CAN BE INPUT ONE OF THREE WAYS:"
60 PRINT "     1. 3 PTS     INDICATOR=1"
70 PRINT "     2. PT, ROT, FB   INDICATOR=2"
80 PRINT "     3. PLANE COEFF   INDICATOR=3"
90 FOR I=1TO 3:SELECT PRINT 005(64)
100 PRINT "PLANE",I:INPUT "INDICATOR=",J
110 IF J=1THEN 120:IF J=2THEN 380:IF J=3THEN 440:PRINT "WRONG INDICATO
R TRY AGAIN":GOTO 100
120 PRINT "INPUT THE 3 PTS":FOR K=1TO 3:PRINT "POINT",K
130 INPUT "X,Y,Z=",X(K),Y(K),Z(K):NEXT K
140 V1=X(1)-X(2) :V2=Y(1)-Y(2):V3=Z(1)-Z(2)
150 V4=X(1)-X(3): V5=Y(1)-Y(3):V6=Z(1)-Z(3)
160 N1=V2*V6-V3*V5:N2=-V1*V6+V3*V4:N3=V1*V5-V2*V4
170 T1=SQR(N1↑2+N2↑2+N3↑2)
180 IF T1[]0THEN 190:PRINT "PTS ARE COLLINEAR, TRY AGAIN":GOTO 120
190 F=ARCSIN(N3/T1)
200 IF COS(F)[]0THEN 210:I6=1:IF N3[0THEN 250:I6=2:GOTO 250
210 IF N1/(T1*COS(F))]1THEN 220:IF N1/(T1*COS(F))[-1THEN 230:GOTO 240
220 R=0:GOTO 250
230 R=180:GOTO 250
240 R=ARCCOS(N1/(T1*COS(F))):IF N2]=0THEN 250:R=360-R
250 A(I,1)=N1/T1:A(I,2)=N2/T1:A(I,3)=N3/T1:A(I,4)=A(I,1)*X(1)+A(I,2)*Y
(1)+A(I,3)*Z(1)
260 SELECT PRINT 215(80):PRINT "INPUT FOR PLANE",I
270 IF I6]0THEN 300
280 PRINTUSING 290,X(1),Y(1),Z(1),R,F:GOTO 320
290% X=-#####.####  Y=-#####.####  Z=-#####.####  ROT=-###.###  F B=-
###.###
300 PRINTUSING 310,X(1),Y(1),Z(1),F
310% X=-#####.####  Y=-#####.####  Z=-#####.####  ROT=*****  FB=-###.
###
320 FOR L=2TO 3:PRINTUSING 330,X(L),Y(L),Z(L):NEXT L
330%    -#####.####    -#####.####    -#####.####
340 IF I6=0THEN 350:IF I6=1THEN 345:PRINT "*****SINCE FB=90 ROT NOT UN
IQUE":I6=0:GOTO 350
345 I6=0:PRINT "*****SINCE FB=-90 ROT NOT UNIQUE"
350 PRINTUSING 360,A(I,1),A(I,2),A(I,3),A(I,4)
360%COEFFICIENTS   A=-#.#####  B=-#.#####  C=-#.#####  D=-#####.####
370 GOTO 480
380 PRINT "INPUT PT, ROT, F B":INPUT "X,Y,Z,ROT,FB=",X(I),Y(I),Z(I),R,
F
390 A(I,1)=COS(F)*COS(R):A(I,2)=COS(F)*SIN(R):A(I,3)=SIN(F)
```

Table A-VII. PLANEINT Program Listing (Continued)

```
400 A(I,4)=A(I,1)*X(I)+A(I,2)*Y(I)+A(I,3)*Z(I)
410 SELECT PRINT 215(80):PRINT "INPUT FOR PLANE",I
420 PRINTUSING 290,X(I),Y(I),Z(I),R,F
430 PRINTUSING 360,A(I,1),A(I,2),A(I,3),A(I,4):GOTO 480
440 INPUT "A,B,C,D=",A(I,1),A(I,2),A(I,3),A(I,4)
450 SELECT PRINT 215(80):PRINT "INPUT FOR PLANE",I
460 PRINTUSING 470,A(I,1),A(I,2),A(I,3),A(I,4):GOTO 480
470%COEFFICIENTS   A=-#####.#####  B=-#####.#####  C=-#####.#####  D=-
#####.#####
480 NEXT I
490 D=A(1,1)*(A(2,2)*A(3,3)-A(3,2)*A(2,3))-A(1,2)*(A(2,1)*A(3,3)-A(2,3
)*A(3,1))+A(1,3)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
500 SELECT PRINT 215(80)
510 IF D[]OTHEN 520:PRINT "PLANES DO NOT INTERSECT":GOTO 580
520 D1=A(1,4)*(A(2,2)*A(3,3)-A(3,2)*A(2,3))-A(1,2)*(A(2,4)*A(3,3)-A(2,
3)*A(3,4))+A(1,3)*(A(2,4)*A(3,2)-A(3,4)*A(2,2))
530 D2=A(1,1)*(A(2,4)*A(3,3)-A(3,4)*A(2,3))-A(1,4)*(A(2,1)*A(3,3)-A(2,
3)*A(3,1))+A(1,3)*(A(2,1)*A(3,4)-A(3,1)*A(2,4))
540 D3=A(1,1)*(A(2,2)*A(3,4)-A(3,2)*A(2,4))-A(1,2)*(A(2,1)*A(3,4)-A(2,
4)*A(3,1))+A(1,4)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
550 X=D1/D:Y=D2/D:Z=D3/D
560 PRINTUSING 570,X,Y,Z
570%INTERSECTION POINT   X=-#####.####  Y=-#####.####  Z=-#####.####
580 PRINT :PRINT :SELECT PRINT 005(64)
590 PRINT "TO STOP, KEY CLEAR CR-LF"
600 PRINT "TO RUN AGAIN, KEY CONTINUE CR-LF"
610 STOP :GOTO 90
620 STOP :END
```

Table A-VIII. LINEPLAN Program Listing

```
10 DIM X(10),Y(10),Z(10)
20 SELECT D :SELECT PRINT 005(64)
30 PRINT "LINEPLAN PROGRAM":PRINT "FINDS INTERSECTION OF A LINE AND PL
ANE"
40 PRINT "LINE IS INPUT BY A POINT AND 3 DELTAS"
50 PRINT "PLANE CAN BE INPUT ONE OF THREE WAYS:"
60 PRINT "     1. 3 PTS     INDICATOR=1"
70 PRINT "     2. PT, ROT, FB   INDICATOR=2"
80 PRINT "     3. PLANE COEFF   INDICATOR=3"
90 SELECT PRINT 005(64):PRINT "INPUT THE LINE":INPUT "X,Y,Z=",X(4),Y(4
),Z(4)
100 INPUT "DEL X, DEL Y, DEL Z=",D1,D2,D3
110 IF D1[]0THEN 130:IF D2[]0THEN 130:IF D3[]0THEN 130
120 PRINT "ERROR  DELTAS ALL ZERO, TRY AGAIN":GOTO 90
130 INPUT "INDICATOR FOR PLANE INPUT=",J
140 IF J=1THEN 150:IF J=2THEN 410:IF J=3THEN 470:PRINT "WRONG INDICATO
R TRY AGAIN":GOTO 100
150 PRINT "INPUT THE 3 PTS":FOR K=1TO 3:PRINT "POINT",K
160 INPUT "X,Y,Z=",X(K),Y(K),Z(K):NEXT K
170 V1=X(1)-X(2) :V2=Y(1)-Y(2):V3=Z(1)-Z(2)
180 V4=X(1)-X(3): V5=Y(1)-Y(3):V6=Z(1)-Z(3)
190 N1=V2*V6-V3*V5:N2=-V1*V6+V3*V4:N3=V1*V5-V2*V4
200 T1=SQR(N1!2+N2!2+N3!2)
210 IF T1[]0THEN 220:PRINT "PTS ARE COLLINEAR, TRY AGAIN":GOTO 150
220 F=ARCSIN(N3/T1)
230 IF COS(F)[]0THEN 240:I6=1:IF N3[0THEN 280:I6=2:GOTO 280
240 IF N1/(T1*COS(F))]1THEN 250:IF N1/(T1*COS(F))[-1THEN 260:GOTO 270
250 R=0:GOTO 280
260 R=180:GOTO 280
270 R=ARCCOS(N1/(T1*COS(F))):IF N2]=0THEN 280:R=360-R
280 A=N1/T1:B=N2/T1:C=N3/T1:D=A*X(1)+B*Y(1)+C*Z(1)
290 SELECT PRINT 215(80):PRINT "INPUT FOR PLANE"
300 IF I6]0THEN 330
310 PRINTUSING 320,X(1),Y(1),Z(1),R,F:GOTO 350
320%  X=-#####.####  Y=-#####.####  Z=-#####.####  ROT=-###.###  F B=-
###.###
330 PRINTUSING 340,X(1),Y(1),Z(1),F
340%  X=-#####.####  Y=-#####.####  Z=-#####.####  ROT=*****  FB=-###.
###
350 FOR L=2TO 3:PRINTUSING 360,X(L),Y(L),Z(L):NEXT L
360%    -#####.####    -#####.####    -#####.####
370 IF I6=0THEN 380:IF I6=1THEN 375:PRINT "*****SINCE FB=90 ROT NOT UN
IQUE":I6=0:GOTO 380
375 PRINT "*****SINCE FB=-90 ROT NOT UNIQUE":I6=0
380 PRINTUSING 390,A,B,C,D
390%COEFFICIENTS   A=-#.#####  B=-#.#####  C=-#.#####  D=-#####.####
400 GOTO 520
```

Table A-VIII. LINEPLAN Program Listing (Continued)

```
410 PRINT "INPUT PT, ROT, F B":INPUT "X,Y,Z,ROT,FB=",X(1),Y(1),Z(1),R,
F
420 A=COS(F)*COS(R):B=COS(F)*SIN(R):C=SIN(F)
430 D=A*X(1)+B*Y(1)+C*Z(1)
440 SELECT PRINT 215(80):PRINT "INPUT FOR PLANE"
450 PRINTUSING 320,X(1),Y(1),Z(1),R,F
460 PRINTUSING 390,A,B,C,D:GOTO 520
470 INPUT "A,B,C,D=",A,B,C,D
480 IF A[]OTHEN 490:IF B[]OTHEN 490:IF C[]OTHEN 490:PRINT "ERROR  ALL
COEFFICIENTS = 0, TRY AGAIN":GOTO 470
490 SELECT PRINT 215(80):PRINT "INPUT FOR PLANE"
500 PRINTUSING 510,A,B,C,D
510%COEFFICIENTS   A=-#####.#####  B=-#####.#####  C=-#####.#####  D=-
#####.#####
520 SELECT PRINT 215(80):PRINT :PRINT "LINE INPUT"
530 PRINTUSING 540,X(4),Y(4),Z(4):PRINTUSING 550,D1,D2,D3
540%X=-#####.####  Y=-#####.####  Z=-#####.####
550%DEL X=-#####.####  DEL Y=-#####.####  DEL Z=-#####.####
560 IF A*D1+B*D2+C*D3[]OTHEN 570:PRINT "LINE PARALLEL TO PLANE, NO INT
ERSECTION":GOTO 610
570 S=(D-A*X(4)-B*Y(4)-C*Z(4))/(A*D1+B*D2+C*D3)
580 X=D1*S+X(4):Y=D2*S+Y(4):Z=D3*S+Z(4)
590 PRINT :PRINTUSING 600,X,Y,Z
600%INTERSECTION POINT     X=-#####.####  Y=-#####.####  Z=-#####.####
610 PRINT :PRINT :PRINT :SELECT PRINT 005(64)
620 PRINT "TO STOP, KEY CLEAR CR-LF"
630 PRINT "TO RUN AGAIN, KEY CONTINUE CR-LF"
640 STOP :GOTO 40
650 STOP :END
```

Table A-IX. LINELINE Program Listing

```
10 PRINT "LINE LINE PROGRAM---FINDS INTERSECTION POINT OF TWO LINES"
20 PRINT
30 PRINT "LINE 1:"
40 INPUT "X,Y,DEL X,DEL Y=",X1,Y1,D1,D2:PRINT "LINE 2:": INPUT "X,Y,DE
L X,DEL Y=",X2,Y2,D3,D4
50 SELECT PRINT 215(80)
60 PRINT "LINE        X             Y          DEL X        DEL Y"
70 J=1:PRINTUSING 90,J,X1,Y1,D1,D2
80 J=2:PRINTUSING 90,J,X2,Y2,D3,D4
90% #   -#####.####  -#####.####  -#####.####  -#####.####
100 IF D2[]0.THEN 110:IF D4[]0.THEN 110:IF Y1=Y2THEN 240:GOTO 260
110 IF D1=0.THEN 210:M1=D2/D1:B1=Y1-M1*X1
120 IF D3=0.THEN 250:M2=D4/D3:B2=Y2-M2*X2
130 IF M1=M2THEN 260:X=(B2-B1)/(M1-M2):Y=M1*X+B1
140 PRINTUSING 150,X,Y
150%INTERSECTION POINT    X=-#####.####  Y=-#####.####
160 PRINT   :PRINT
170 SELECT PRINT 005(64)
180 SELECT PRINT 005(64):PRINT "TO RUN AGAIN, KEY CONTINUE CR-LF"
190 PRINT "TO STOP, KEY CLEAR CR-LF":STOP
200 GOTO 30
210 IF D3=0.THEN 230:X=X1:Y=D4/D3*X+(Y2-D4/D3*X2)
220 GOTO 140
230 IF X1=X2THEN 240:GOTO 260
240 PRINT "LINES ARE SAME":PRINT :PRINT :GOTO 180
250 X=X2:Y=M1*X+B1:GOTO 140
260 PRINT "LINES DO NOT INTERSECT":PRINT :PRINT :GOTO 180
270 STOP
```

Table A-X. RFARB Program Listing

```
10 DIM X(10),Y(10),Z(10)
20 SELECT D
30 SELECT PRINT 005(64):PRINT "RFARB PROGRAM"
40 PRINT "FINDS ARB8 DEFINED BY A POINT, ROT ANGLE,"
50 PRINT "FB ANGLE, 2 COORDINATES OF 3 OTHER POINTS,"
60 PRINT "AND A THICKNESS"
70 INPUT "X,Y,Z OF PT. 1, ROT, FB ANGLES",X(1),Y(1),Z(1),R,F
80 A=COS(F)*COS(R) :B=COS(F)*SIN(R) :C=SIN(F)
90 D=A*X(1)+B*Y(1)+C*Z(1)
100 FOR I=2TO 4
110 PRINT "PT NUM=",I
120 INPUT "INPUT INDICATOR AND 2 KNOWN COORDINATES",I1,I2,I3,U1,U2
130 IF I1]OTHEN 190
140 IF I2]OTHEN 220
150 IF I3=OTHEN 490
160 IF C=OTHEN 500
170 X(I)=U1 :Y(I)=U2:Z(I)=(D-A*U1-B*U2)/C
180 GOTO 240
190 IF A=OTHEN 520
200 Y(I)=U1 :Z(I)=U2 :X(I)=(D-B*U1-C*U2)/A
210 GOTO 240
220 IF B=OTHEN 530
230 X(I)=U1 :Z(I)=U2 :Y(I)=(D-A*U1-C*U2)/B
240 NEXT I
250 INPUT "THICKNESS=",T
260 FOR I=5TO 8
270 J=I-4
280 X(I)=X(J)+(A*T)
290 Y(I)=Y(J)+(B*T)
300 Z(I)=Z(J)+(C*T)
310 NEXT I
320 D1=X(5)*A+Y(5)*B+Z(5)*C
330 SELECT PRINT 215(80)
340 PRINT "PT          X              Y              Z"
350 FOR I=1TO 8
360 PRINTUSING 380,I,X(I),Y(I),Z(I)
370 NEXT I
380 ###   -#####.####  -#####.####  -#####.####
390 PRINT "FACE      A          B          C          D          ROT
    F B"
400 K1=1234 :K2=5678
410 PRINTUSING 420,K1,A,B,C,D,R,F:PRINTUSING 420,K2,A,B,C,D1,R,F
420 ####  -#.#####  -#.#####  -#.#####  -####.#####  -###.##  -###.##
430 PRINT "THICKNESS=",T
440 PRINT :PRINT
450 SELECT PRINT 005(64)
460 PRINT "TO RUN AGAIN, KEY CONTINUE CR-LF"
```

Table A-X. RFARB Program Listing (Continued)

```
470 PRINT "TO STOP, KEY CLEAR CR-LF"
480 STOP :GOTO 70
490 PRINT "ERROR, NO COORDINATE TO SOLVE FOR":PRINT :PRINT :GOTO 110
500 PRINT "Z NOT UNIQUE IN THIS PLANE":PRINT :PRINT
510 GOTO 110
520 PRINT "X NOT UNIQUE IN THIS PLANE" :PRINT :PRINT :GOTO 110
530 PRINT "Y NOT UNIQUE IN THIS PLANE" :PRINT :PRINT :GOTO 110
540 END
```

Table A-XI. 3PTARB Program Listing

```
10 DIM X(10),Y(10),Z(10)
20 SELECT D
30 SELECT PRINT 005(64)
40 PRINT "3PTARB PROGRAM":PRINT "FINDS ARB8 DEFINED BY 3 POINTS, 2 COO
RDINATES OF"
50 PRINT "A 4TH POINT, AND A THICKNESS"
60 PRINT "INPUT 3 POINTS"
70 FOR I=1TO 3:PRINT "POINT",I: INPUT "X=",X(I):INPUT "Y=",Y(I):INPUT
"Z=",Z(I):NEXT I
80 V1=X(1)-X(2):V2=Y(1)-Y(2):V3=Z(1)-Z(2)
90 V4=X(1)-X(3):V5=Y(1)-Y(3):V6=Z(1)-Z(3)
100 N1=V2*V6-V3*V5:N2=-V1*V6+V3*V4:N3=V1*V5-V2*V4
110 T1=SQR(N1!2+N2!2+N3!2)
120 IF T1[]0THEN 130:PRINT "PTS COLLINEAR":GOTO 390
130 F=ARCSIN(N3/T1)
140 IF COS(F)[]0THEN 150:I1=1:IF N3[0THEN 190:I1=2:GOTO 190
150 IF N1/(T1*COS(F))]1THEN 160:IF N1/(T1*COS(F))[-1THEN 170:GOTO 180
160 R=0:GOTO 190
170 R=180:GOTO 190
180 R=ARCCOS(N1/(T1*COS(F))):IF N2]=0THEN 190:R=360-R
190 A=N1/T1:B=N2/T1:C=N3/T1:D=A*X(1)+B*Y(1)+C*Z(1)
200 INPUT "INDICATOR AND 2 KNOWN COORDINATES=",I2,I3,I4,U1,U2
210 IF I2]0THEN 240:IF I3]0THEN 260:IF I4[=0THEN 480
220 IF C=0THEN 490:X(4)=U1:Y(4)=U2:Z(4)=(D-A*U1-B*U2)/C
230 GOTO 270
240 IF A=0THEN 500:Y(4)=U1:Z(4)=U2:X(4)=(D-B*U1-C*U2)/A
250 GOTO 270
260 IF B=0THEN 510:X(4)=U1:Z(4)=U2:Y(4)=(D-A*U1-C*U2)/B
270 INPUT "THICKNESS=",T
280 FOR I=5TO 8:J=I-4:X(I)=X(J)+T*A:Y(I)=Y(J)+T*B
290 Z(I)=Z(J)+T*C:NEXT I:D1=A*X(5)+B*Y(5)+C*Z(5)
300 SELECT PRINT 215(80)
310 PRINT "PT          X              Y              Z"
320 FOR I=1TO 8:PRINTUSING 330,I,X(I),Y(I),Z(I):NEXT I
330%##   -#####.####  -#####.####  -#####.####
340 PRINT "FACE      A          B          C          D          ROT
   FB"
350 K1=1234:K2=5678:IF I1]0THEN 420
360 PRINTUSING 370,K1,A,B,C,D,R,F:PRINTUSING 370,K2,A,B,C,D1,R,F
370%####  -#.#####  -#.#####  -#.#####  -####.#####  -###.##  -###.##
380 PRINT "THICKNESS=",T
390 PRINT :PRINT
400 SELECT PRINT 005(64):PRINT "TO RUN AGAIN, KEY CONTINUE CR-LF"
410 PRINT "TO STOP, KEY CLEAR CR-LF":STOP :GOTO 60
420 SELECT PRINT 215(80)
430 PRINTUSING 440,K1,A,B,C,D,F:PRINTUSING 440,K2,A,B,C,D1,F
440%####  -#.#####  -#.#####  -#.#####  -####.#####  *******  -###.##
```

Table A-XI. 3PTARB Program Listing (Continued)

```
450 IF I1=2THEN 460:I1=0:PRINT "*** NOTE  SINCE FB IS -90 ROT IS NOT U
NIQUE***":GOTO 470
460 PRINT "*** NOTE  SINCE FB IS 90 ROT NOT UNIQUE ***":I1=0
470 PRINT "THICKNESS=",T:PRINT :PRINT :GOTO 400
480 PRINT "INPUT ERROR NO COORDINATE TO SOLVE FOR":PRINT :PRINT :GOTO
400
490 PRINT "Z NOT UNIQUE IN THIS PLANE":PRINT :PRINT :GOTO 400
500 PRINT "X NOT UNIQUE IN THIS PLANE":PRINT :PRINT :GOTO 400
510 PRINT "Y NOT UNIQUE IN THIS PLANE":PRINT :PRINT :GOTO 400
```

Table A-XII. NORMVEC Program Listing

```
10 PRINT "NORMAL VECTOR PROGRAM"
20 PRINT "FINDS VECTOR OF LENGTH L1 IN DIRECTION OF GIVEN"
30 PRINT "VECTOR V=(X,Y) AND VECTOR OF LENGTH L2 NORMAL"
40 PRINT "TO GIVEN VECTOR V"
50 PRINT "ORDER OF INPUT:  X,Y,L1,L2"
60 INPUT "INPUT X,Y,L1,L2",X,Y,L1,L2
70 S=SQR(X!2+Y!2)
80 X1=L1*X/S
90 Y1=L1*Y/S
100 X2=-L2*Y/S
110 Y2=L2*X/S
120 SELECT PRINT 215(80)
130 PRINT "INPUT VECTOR:"
140 PRINTUSING 230,S,X,Y
150 PRINT "VECTOR IN DIRECTION OF INPUT VECTOR:"
160 PRINTUSING 230,L1,X1,Y1
170 PRINT "VECTOR PERPENDICULAR TO INPUT VECTOR:"
180 PRINTUSING 230,L2,X2,Y2
190 PRINT :PRINT
200 SELECT PRINT 005(64)
210 PRINT "TO RUN AGAIN, KEY CONTINUE,CR-LF"
220 PRINT "TO STOP, KEY CLEAR,CR-LF"
230%LENGTH=-#####.###  DEL X=-#####.###  DEL Y=-#####.###
240 STOP
250 GOTO 60
```

Table A-XIII. PERPENV Program Listing

```
10 SELECT PRINT 005(64)
20 SELECT D
30 PRINT "PERPENV PROGRAM"
40 PRINT "FINDS VECTOR PERPENDICULAR TO 2 GIVEN VECTORS"
50 INPUT "VECTOR 1=",V1,V2,V3
60 V4=SQR(V1!2+V2!2+V3!2):IF V4[]OTHEN 80
70 PRINT "***** T R Y   A G A I N *****":GOTO 50
80 INPUT "VECTOR 2=",W1,W2,W3
90 W4=SQR(W1!2+W2!2+W3!2):IF W4[]OTHEN 110
100 PRINT "***** T R Y   A G A I N *****":GOTO 80
110 INPUT "LENGTH OF PERPENDICULAR VECTOR=",L
120 IF L]OTHEN 140
130 PRINT "***** T R Y    A G A I N *****":GOTO 110
140 D=V1*W1+V2*W2+V3*W3
150 T=ARCCOS(D/(V4*W4))
160 U1=V2*W3-V3*W2
170 U2=-(V1*W3-V3*W1)
180 U3=V1*W2-V2*W1
190 U4=SQR(U1!2+U2!2+U3!2)
200 IF U4]OTHEN 220
210 PRINT "ERROR--VECTORS ARE MULTIPLES":GOTO 370
220 U1=L*U1/U4:U2=L*U2/U4:U3=L*U3/U4
230 SELECT PRINT 215(90)
240 PRINT "INPUT VECTORS:"
250 PRINTUSING 260,V1,V2,V3
260 %DEL X=-#####.####   DEL Y=-#####.####   DEL Z=-#####.####
270 PRINTUSING 260,W1,W2,W3
280 PRINTUSING 290,V4,W4
290 %LENGTHS     VECTOR 1=-####.####   VECTOR 2=-####.####
300 PRINTUSING 310,T:PRINT
310 %ANGLE BETWEEN INPUT VECTORS=-###.### DEGREES
320 PRINT "VECTOR PERPENDICULAR TO INPUT VECTORS:"
330 PRINTUSING 350,U1,U2,U3
340 PRINTUSING 360,-U1,-U2,-U3
350 %     DX=-#####.####   DY=-#####.####   DZ=-#####.####
360 % OR  DX=-#####.####   DY=-#####.####   DZ=-#####.####
370 PRINT :PRINT :PRINT :SELECT PRINT 005(64):GOTO 50
```

Table A-XIV. AMTRACK Program Listing

```
10 DIM O(120),R$(12)
20 FOR I=1TO 12:READ R$(I):NEXT I
30 DATA "RPP ","BOX ","ARB8","RCC ","FRNT SLOPE","TRACK BOT ","REAR SL
OPE","TRACK TOP ","TRCK IDLER","TRCK DRIVE","IDLER DUM ","DRIVE DUM "
40 SELECT PRINT 005(64)
50 PRINT "AMTRACK PROGRAM"
60 PRINT "ADDS TRACKS TO DOMESTIC VEHICLES"
70 PRINT "ASSUMES  1. IDLER WHEEL IS IN FRONT OF FIRST ROAD WHEEL"
80 PRINT "         2. DRIVE WHEEL IS TO REAR OF LAST ROAD WHEEL"
90 PRINT "         3. Z COORDINATE OF THE ROAD WHEELS IS EQUAL"
100 INPUT "X,Z OF FIRST ROAD WHEEL=",X1,Z1
110 INPUT "X,Z OF LAST ROAD WHEEL=",X3,Z3
120 IF Z1=Z3THEN 130:PRINT "*** ERROR - Z'S OF THE ROAD WHEELS MUST BE
 THE SAME ***":GOTO 100
130 IF X1]X3THEN 140:PRINT "*** ERROR - LAST ROAD WHEEL TO THE FRONT O
F FIRST ROAD WHEEL ***":GOTO 100
140 INPUT "RADIUS OF ROAD WHEELS=",R1
150 IF R1]OTHEN 160:PRINT "*** ERROR - RADIUS [= 0 ***":GOTO 140
160 INPUT "X,Z OF IDLER WHEEL=",X2,Z2
170 IF X2]X1THEN 180:PRINT "*** ERROR - IDLER WHEEL IS CONSIDERED AS I
N FRONT OF 1ST ROAD WHEEL ***":GOTO 160
180 INPUT "RADIUS OF IDLER WHEEL=",R2
190 IF R2]OTHEN 200:PRINT "*** ERROR - RADIUS [= 0 ***":GOTO 180
200 INPUT "X,Z OF DRIVE WHEEL=",X4,Z4
210 IF X4[X3THEN 220:PRINT "*** ERROR - DRIVE WHEEL IS CONSIDERED AS T
O REAR OF LAST ROAD WHEEL ***":GOTO 200
220 INPUT "RADIUS OF DRIVE WHEEL=",R4
230 IF R4]OTHEN 240:PRINT "*** ERROR - RADIUS [= 0 ***":GOTO 220
240 INPUT "YMIN, YMAX OF TRACK=",Y,Y1
250 IF Y[Y1THEN 260:PRINT "*** ERROR - YMIN ]= YMAX ***":GOTO 240
260 INPUT "THICKNESS OF TRACK=",T
270 IF T]OTHEN 280:PRINT "*** ERROR - TRACK THICKNESS MUST BE ]0 ***":
GOTO 260
280 D7=4
290 X=X1:Z=Z1:R=R1:K9=0:D=1
300 M=X2:N=Z2:Q=R2
310 A1=X:B1=Z-R
320 A1=A1+D7
330 GOSUB '01
340 GOSUB '02
350 IF D3]=0THEN 320
360 C3=(B2-B1)/SQR((B1-B2)!2+(A1-A2)!2)
370 C4=(A2-A1)/SQR((B1-B2)!2+(A1-A2)!2)
380 C5=D*C3:C6=-C4*D
390 B4=B1-T:A4=(A1+(C5*T))-(((B1+(T*C6))-B4)*C4/C3)
400 O(1+K9)=A4:O(2+K9)=Y:O(3+K9)=B4
410 O(4+K9)=A2+(C5*T)-A4:O(5+K9)=0:O(6+K9)=B2+(C6*T)-B4
```

Table A-XIV. AMTRACK Program Listing (Continued)

```
420 O(7+K9)=-C5*T:O(8+K9)=0:O(9+K9)=-C6*T
430 O(10+K9)=0:O(11+K9)=Y1-Y:O(12+K9)=0
440 IF K9]OTHEN 470
450 O(26)=A4:O(27)=Y:O(28)=Y1:O(29)=B1-T:O(30)=B1
460 K9=12:X=X3:Z=Z3:D=-1:M=X4:N=Z4:Q=R4:D7=-D7:GOTO 310
470 O(25)=A4
480 C1=(O(4)+O(1)-X2)/SQR((O(4)+O(1)-X2)!2+(O(6)+O(3)-Z2)!2)
490 C2=(O(6)+O(3)-Z2)/SQR((O(4)+O(1)-X2)!2+(O(6)+O(3)-Z2)!2)
500 X=X2:R=R2:Z=Z2:K9=0
510 O(34+K9)=X+((R+T+300)*C1)
520 O(35+K9)=Y1+10
530 O(36+K9)=Z+((R+T+300)*C2)
540 O(31+K9)=O(34+K9):O(32+K9)=O(35+K9)
550 O(33+K9)=Z+R+T+50
560 O(37+K9)=X:O(38+K9)=O(35+K9):O(39+K9)=Z
570 O(40+K9)=X:O(41+K9)=O(35+K9):O(42+K9)=O(33+K9)
580 O(43+K9)=O(31+K9):O(44+K9)=Y-10:O(45+K9)=O(33+K9)
590 O(46+K9)=O(34+K9):O(47+K9)=Y-10:O(48+K9)=O(36+K9)
600 O(49+K9)=O(37+K9):O(50+K9)=Y-10:O(51+K9)=O(39+K9)
610 O(52+K9)=O(40+K9):O(53+K9)=Y-10:O(54+K9)=O(42+K9)
620 IF K9]OTHEN 660:K9=24:X=X4:Z=Z4:R=R4
630 C1=(O(16)+O(13)-X)/SQR((O(16)+O(13)-X)!2+(O(18)+O(15)-Z)!2)
640 C2=(O(18)+O(15)-Z)/SQR((O(16)+O(13)-X)!2+(O(18)+O(15)-Z)!2)
650 GOTO 510
660 K9=0:X=X2:Z=Z2:R=R2+T
670 O(79+K9)=X:O(80+K9)=Y:O(81+K9)=Z
680 O(82+K9)=0:O(83+K9)=Y1-Y:O(84+K9)=0:O(85+K9)=R
690 IF K9]OTHEN 700:K9=7:X=X4:Z=Z4:R=R4+T:GOTO 670
700 IF Z2+R2[]Z4+R4 THEN 730
710 O(94)=X2+10:O(93)=X4-10:O(95)=Y:O(96)=Y1
720 O(97)=Z2+R2:O(98)=O(97)+T:L7=98:GOTO 790
730 C1=(X4-X2)/SQR((X2-X4)!2+(Z2+R2-Z4-R4)!2)
740 C2=(Z4+R4-Z2-R2)/SQR((X2-X4)!2+(Z2+R2-Z4-R4)!2)
750 O(93)=X2-(10*C1):O(94)=Y1:O(95)=Z2+R2-(10*C2)
760 O(96)=(X4-X2)+(20*C1):O(97)=0:O(98)=(Z4+R4-Z2-R2)+(20*C2)
770 O(99)=T*C2:O(100)=0:O(101)=-T*C1
780 O(102)=0:O(103)=-(Y1-Y):O(104)=0:L7=104
790 K9=0:X=X2:Z=Z2:R=R2
800 O(L7+K9+1)=X:O(L7+K9+2)=Y:O(L7+K9+3)=Z
810 O(L7+K9+4)=0:O(L7+K9+5)=Y1-Y:O(L7+K9+6)=0
820 O(L7+K9+7)=R
830 IF K9]OTHEN 850
840 K9=7:X=X4:Z=Z4:R=R4:GOTO 800
850 SELECT PRINT 215(100):I=1:PRINTUSING 860,I,R$(2),O(1),O(2),O(3),O(
4),O(5),O(6),R$(5)
860 ### #### -#####.#### -#####.#### -#####.#### -#####.#### -#####.##
## -#####.####    ##########
```

Table A-XIV. AMTRACK Program Listing (Continued)

```
870 PRINTUSING 880,0(7),0(8),0(9),0(10),0(11),0(12)
880 %        -#####.#### -#####.#### -#####.#### -#####.#### -#####.##
## -#####.####
890 I=2:PRINTUSING 860,I,R$(2),0(13),0(14),0(15),0(16),0(17),0(18),R$(
7)
900 PRINTUSING 880,0(19),0(20),0(21),0(22),0(23),0(24)
910 I=3:PRINTUSING 860,I,R$(1),0(25),0(26),0(27),0(28),0(29),0(30),R$(
6)
920 I=4:PRINTUSING 860,I,R$(3),0(31),0(32),0(33),0(34),0(35),0(36),R$(
11)
930 PRINTUSING 880,0(37),0(38),0(39),0(40),0(41),0(42)
940 PRINTUSING 880,0(43),0(44),0(45),0(46),0(47),0(48)
950 PRINTUSING 880,0(49),0(50),0(51),0(52),0(53),0(54)
960 I=5:PRINTUSING 860,I,R$(3),0(55),0(56),0(57),0(58),0(59),0(60),R$(
12)
970 PRINTUSING 880,0(61),0(62),0(63),0(64),0(65),0(66)
980 PRINTUSING 880,0(67),0(68),0(69),0(70),0(71),0(72)
990 PRINTUSING 880,0(73),0(74),0(75),0(76),0(77),0(78)
1000 I=6:PRINTUSING 860,I,R$(4),0(79),0(80),0(81),0(82),0(83),0(84),R$
(9)
1010 PRINTUSING 880,0(85)
1020 I=7:PRINTUSING 860,I,R$(4),0(86),0(87),0(88),0(89),0(90),0(91),R$
(10)
1030 PRINTUSING 880,0(92)
1040 I=8:IF Z2+R2=Z4+R4 THEN 1070
1050 PRINTUSING 860,I,R$(2),0(93),0(94),0(95),0(96),0(97),0(98),R$(8)
1060 PRINTUSING 880,0(99),0(100),0(101),0(102),0(103),0(104):GOTO 1080
1070 PRINTUSING 860,I,R$(1),0(93),0(94),0(95),0(96),0(97),0(98),R$(8)
1080 I=9:PRINTUSING 860,I,R$(4),0(L7+1),0(L7+2),0(L7+3),0(L7+4),0(L7+5
),0(L7+6),R$(11)
1090 PRINTUSING 880,0(L7+7)
1100 I=10:PRINTUSING 860,I,R$(4),0(L7+8),0(L7+9),0(L7+10),0(L7+11),0(L
7+12),0(L7+13),R$(12)
1110 PRINTUSING 880,0(L7+14)
1120 PRINT :PRINT :PRINT "REGION TABLE"
1130 PRINTUSING 1140,1,1,-4,0,R$(5)
1140 % ##   -###  -###  -###          ##########
1150 PRINTUSING 1140,2,2,-5,0,R$(7)
1160 PRINTUSING 1140,3,3,-1,-2,R$(6)
1170 PRINTUSING 1140,4,6,4,-9,R$(9)
1180 PRINTUSING 1140,5,7,5,-10,R$(10)
1190 PRINTUSING 1140,6,8,-4,-5,R$(8)
1200 GOTO 40
1210 DEFFN'01
1220 K2=(A1-N)!2+(B1-N)!2
1230 G1=M+(((Q!2)*(A1-N))-(Q*(B1-N)*SQR(K2-Q!2)))/K2
1240 G2=M+(((Q!2)*(A1-N))+(Q*(B1-N)*SQR(K2-Q!2)))/K2
```

Table A-XIV. AMTRACK Program Listing (Continued)

```
1250 G3=N+(((Q!2)*(B1-N))+(Q*(A1-M)*SQR(K2-Q!2)))/K2
1260 G4=N+(((Q!2)*(B1-N))-(Q*(A1-M)*SQR(K2-Q!2)))/K2
1270 IF G4[G3THEN 1280:B2=G3:A2=G1:GOTO 1290
1280 B2=G4:A2=G2
1290 RETURN
1300 DEFFN'02
1310 D1=A2-A1:D2=B2-B1
1320 S=D2/D1:Y7=B2-S*A2:A=1+S!2
1330 B=2*S*(Y7-Z)-2*X:C=X!2+(Y7-Z)!2-R!2
1340 D3=B!2-4*A*C
1350 RETURN
```

Table A-XV. SOLIDROT Program Listing

```
10 DIM A$(20),D(25),B(4,6),S(25),R(25),W(6)
20 PRINT "SOLIDROT PROGRAM"
30 PRINT "ROTATES IN THE XY, XZ, YZ PLANES SELECTED":PRINT "COM-GEOM S
OLIDS ABOUT ANY POINT"
40 PRINT "POSITIVE ROTATION IS FROM POSITIVE AXIS TO POSITIVE AXIS"
50 FOR I=1TO 20:READ A$(I):NEXT I
60 DATA "RPP ","BOX ","RAW ","SPH ","RCC ","REC ","TRC ","TEC ","TOR "
,"ELL1","ELL ","ARB8","ARB7","ARB6","ARB5","ARB4","BLK","XY","XZ","YZ"
70 SELECT D
80 SELECT PRINT 005(64):INPUT "PLANE OF ROTATION(1=XY,2=XZ,3=Y7)",P
90 IF P[1THEN 100:IF P]3THEN 100:GOTO 110
100 PRINT "*****TRY AGAIN*****":GOTO 80
110 INPUT "ANGLE OF ROTATION(DEG)=",A
120 INPUT "X,Y,Z OF PT TO ROTATE SOLID ABOUT=",X5,Y5,Z5
130 PRINT "THE FOLLOWING SOLID TYPES ARE AVAILABLE"
140 PRINT TAB(4);"RPP(1), BOX(2), RAW(3), SPH(4), RCC(5), REC(6), TRC(
7)"
150 PRINT TAB(4);"TEC(8), TOR(9), ELL1(10), ELL(11), ARB8(12), ARB7(13
)"
160 PRINT TAB(4);"ARB6(14), ARB5(15), ARB4(16)"
170 INPUT "YOUR SOLID TYPE IS",T
180 IF T[0THEN 190:ON T GOTO 210,340,480,550,580,620,680,730,790,830,8
70,910,960,980,1000,1020
190 PRINT "******TRY AGAIN******":GOTO 170
200 REM              ***SOLID TYPE IS RPP***
210 INPUT "XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX=",D(1),D(2),D(3),D(4),D(5),D(
6)
220 FOR I=1TO 5STEP 2:IF D(I)]D(I+1)THEN 230:NEXT I:GOTO 260
230 IF I=1THEN 240:IF I=3THEN 250:PRINT "ERROR  ZMIN]ZMAX, TRY AGAIN":
GOTO 210
240 PRINT "ERROR  XMIN]XMAX, TRY AGAIN":GOTO 210
250 PRINT "ERROR  YMIN]YMAX, TRY AGAIN":GOTO 210
260 FOR I=1TO 6:W(I)=D(I):NEXT I
270 K2=D(3):K3=D(5)
280 H1=D(2)-D(1):W1=D(4)-D(3):D1=D(6)-D(5)
290 D(2)=K2:D(3)=K3
300 D(4)=H1:D(5)=0:D(6)=0
310 D(7)=0:D(8)=W1:D(9)=0
320 D(10)=0:D(11)=0:D(12)=D1:F6=6:F=10:F7=3:GOTO 390
330 REM              ***SOLID TYPE IS BOX***
340 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
350 PRINT "NOTE---THE ORDER OF FOLLOWING VECTORS MAY VARY"
360 INPUT "HEIGHT VECTOR=",D(4),D(5),D(6)
370 INPUT "WIDTH VECTOR=",D(7),D(8),D(9)
380 INPUT "DEPTH VECTOR=",D(10),D(11),D(12):F=10:F7=3:F6=12
390 FOR I=1TO 25:S(I)=D(I):NEXT I
400 FOR I=1TO FSTEP 3:IF I]F7THEN 410:S(I)=D(I)-X5:S(I+1)=D(I+1)-Y5:S(
I+2)=D(I+2)-Z5
```

Table A-XV. SOLIDROT Program Listing (Continued)

```
410 IF P=1THEN 420:IF P=2THEN 430:IF P=3THEN 440:PRINT "INPUT ERROR, T
RY AGAIN":GOTO 80
420 GOSUB '01(S(I),S(I+1),S(I+2)):GOTO 450
430 GOSUB '02(S(I),S(I+1),S(I+2)):GOTO 450
440 GOSUB '03(S(I),S(I+1),S(I+2))
450 IF I]F7THEN 460:R(I)=R(I)+X5:R(I+1)=R(I+1)+Y5:R(I+2)=R(I+2)+Z5
460 NEXT I:GOTO 1030
470 REM              ***SOLID TYPE IS RAW***
480 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
490 PRINT "NOTE-HEIGHT AND WIDTH VECTORS MAY BE INTERCHANGED"
500 INPUT "HEIGHT VECTOR=",D(4),D(5),D(6)
510 INPUT "WIDTH VECTOR=",D(7),D(8),D(9)
520 INPUT "DEPTH VECTOR=",D(10),D(11),D(12)
530 F=10:FG=12:F7=3:GOTO 390
540 REM              ***SOLID TYPE IS SPH***
550 INPUT "X,Y,Z OF CENTER=",D(1),D(2),D(3)
560 INPUT "RADIUS=",D(4):R(4)=D(4):F=1:FG=4:F7=3:GOTO 390
570 REM              ***SOLID TYPE IS RCC***
580 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
590 INPUT "HEIGHT VECTOR=",D(4),D(5),D(6)
600 INPUT "RADIUS OF BASE=",R(7):F=4:FG=7:F7=3:D(7)=R(7):GOTO 390
610 REM              ***SOLID TYPE IS REC***
620 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
630 INPUT "HEIGHT VECTOR=",D(4),D(5),D(6)
640 INPUT "VECTOR DEFINING SEMI-MAJOR AXIS=",D(7),D(8),D(9)
650 INPUT "VECTOR DEFINING SEMI-MINOR AXIS=",D(10),D(11),D(12)
660 F=10:FG=12:F7=3:GOTO 390
670 REM              ***SOLID TYPE IS TRC***
680 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
690 INPUT "HEIGHT VECTOR=",D(4),D(5),D(6)
700 INPUT "LARGE RADIUS=",R(7):INPUT "SMALL RADIUS=",R(8)
710 F=4:FG=8:F7=3:D(7)=R(7):D(8)=R(8):GOTO 390
720 REM              ***SOLID TYPE IS TEC***
730 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
740 INPUT "HEIGHT VECTOR=",D(4),D(5),D(6)
750 INPUT "VECTOR DEFINING SEMI-MAJOR AXIS=",D(7),D(8),D(9)
760 INPUT "VECTOR DEFINING SEMI-MINOR AXIS=",D(10),D(11),D(12)
770 INPUT "RATIO=",R(13):F=10:FG=13:F7=3:D(13)=R(13):GOTO 390
780 REM              ***SOLID TYPE IS TOR***
790 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
800 INPUT "NORMAL VECTOR=",D(4),D(5),D(6)
810 INPUT "R1=",R(7):INPUT "R2=",R(8):F=4:FG=8:F7=3:D(7)=R(7):D(8)=R(8
):GOTO 390
820 REM              ***SOLID TYPE IS ELL1***
830 INPUT "X,Y,Z OF VERTEX=",D(1),D(2),D(3)
840 INPUT "VECTOR DEFINING SEMI-MAJOR AXIS=",D(4),D(5),D(6)
850 INPUT "RADIUS=",R(7):F=4:FG=7:F7=3:D(7)=R(7):GOTO 390
```

Table A-XV. SOLIDROT Program Listing (Continued)

```
860 REM              ***SOLID TYPE IS ELL***
870 INPUT "X,Y,Z OF FOCI 1=",D(1),D(2),D(3)
880 INPUT "X,Y,Z OF FOCI 2=",D(4),D(5),D(6)
890 INPUT "LENGTH OF MAJOR AXIS=",R(7):F=4:F6=7:F7=6:D(7)=R(7):GOTO 39
0
900 REM              ***SOLID TYPE IS ARB8***
910 F1=22:F=22:F6=24:F7=F6
920 J=0:FOR I=1TO F1STEP 3:J=J+1:PRINT "POINT",J
930 INPUT "X,Y,Z=",D(I),D(I+1),D(I+2):NEXT I
940 GOTO 390
950 REM              ***SOLID TYPE IS ARB7***
960 F=19:F1=19:F6=21:F7=F6:GOTO 920
970 REM              ***SOLID TYPE IS ARB6***
980 F=16:F1=16:F6=18:F7=F6:GOTO 920
990 REM              ***SOLID TYPE IS ARB5***
1000 F=13:F1=13:F6=15:F7=F6:GOTO 920
1010 REM              ***SOLID TYPE IS ARB4***
1020 F=10:F1=10:F6=12:F7=F6:GOTO 920
1030 SELECT PRINT 215(85):PRINTUSING 1040,A,A$(P+17)
1040 %ANGLE OF ROTATION=-###.#### DEG IN THE  ##  PLANE
1050 PRINTUSING 1060,X5,Y5,Z5:G6=F6
1060 %PT AROUND WHICH SOLID WAS ROTATED  X=-####.#### Y=-####.#### Z=-
####.####
1070 K=0:FOR I=1TO 4:FOR J=1TO 6:K=K+1:B(I,J)=D(K):NEXT J:NEXT I
1080 IF T[]1THEN 1090:FOR I=1TO 6:B(1,I)=W(I):NEXT I
1090 PRINT "INPUT SOLID":IF T[]4THEN 1100:PRINTUSING 1110,A$(T),B(1,1)
,B(1,2),B(1,3),B(1,4):GOTO 1200
1100 PRINTUSING 1110,A$(T),B(1,1),B(1,2),B(1,3),B(1,4),B(1,5),B(1,6)
1110 %####  -###.#### -###.#### -###.#### -###.#### -####.#### -##
##.####
1120 %      -###.#### -###.#### -####.#### -###.#### -###.#### -##
##.####
1130 G6=G6-6:IF G6[6THEN 1150:FOR I=2TO 4:PRINTUSING 1120,B(I,1),B(I,2
),B(I,3),B(I,4),B(I,5),B(I,6)
1140 G6=G6-6:IF G6[6THEN 1160:NEXT I:GOTO 1200
1150 I=1
1160 IF G6=1THEN 1170:IF G6=2THEN 1180:IF G6=3THEN 1190:IF G6=0THEN 12
00
1170 PRINTUSING 1120,B(I+1,1):GOTO 1200
1180 PRINTUSING 1120,B(I+1,1),B(I+1,2):GOTO 1200
1190 PRINTUSING 1120,B(I+1,1),B(I+1,2),B(I+1,3):GOTO 1200
1200 IF E9=0THEN 1210:E9=0:PRINT :PRINT :GOTO 1260
1210 PRINT "ROTATED SOLID"
1220 IF T[]4THEN 1230:PRINTUSING 1110,A$(T),R(1),R(2),R(3),R(4):PRINT
:PRINT :GOTO 1260
1230 K=0:FOR I=1TO 4:FOR J=1TO 6:K=K+1:B(I,J)=R(K):NEXT J:NEXT I
1240 IF T[]1THEN 1250:T=2:F6=12
```

Table A-XV. SOLIDROT Program Listing (Continued)

```
1250 E9=1:G6=FG:GOTO 1100
1260 SELECT PRINT 005(64)
1270 PRINT "TO RUN AGAIN, KEY CONTINUE  EXECUTE"
1280 PRINT "TO STOP, KEY CLEAR  EXECUTE"
1290 STOP :GOTO 80
1300 DEFFN'01(S1,S2,S3)
1310 R(I)=S1*COS(A)-S2*SIN(A)
1320 R(I+1)=S1*SIN(A)+S2*COS(A)
1330 R(I+2)=S3
1340 RETURN
1350 DEFFN'02(S4,S5,S6)
1360 R(I)=S4*COS(A)-S6*SIN(A)
1370 R(I+1)=S5
1380 R(I+2)=S4*SIN(A)+S6*COS(A)
1390 RETURN
1400 DEFFN'03(S7,S8,S9)
1410 R(I)=S7
1420 R(I+1)=S8*COS(A)-S9*SIN(A)
1430 R(I+2)=S8*SIN(A)+S9*COS(A)
1440 RETURN
```

Table A-XVI. PLOTSOL Program Listing

```
10 DIM A(120),B(120),C(120),D(120),H(120),F(8),G(8),M(8),K8(30)
20 DIM N(240),O(240),P(240),Q(240),R(240)
30 DIM S(120),T(120),U(120),V(120),X(5),Y(5),Z(5)
40 PRINT "PLOTSOL PROGRAM"
50 PRINT "PLOTS SELECTED COM-GEOM SOLIDS AT ANY ASPECT"
60 PRINT "HAS SCALE ADJUST CAPABILITIES"
70 PRINT "SUM OF RCC'S AND TRC'S MUST BE [ = 5"
80 PRINT "MAX NUM OF SPHERES = 5"
90 SELECT D
100 INPUT "NUM OF SOLIDS TO PLOT=",K9:G9=K9
110 IF K9]0THEN 120:PRINT "*** TRY AGAIN ***":PRINT HEX(07):GOTO 100
120 INPUT "AZ, EL FOR THIS VIEW=",A,E
130 X8=0:X9=0:Z8=0:Z9=0:K1=1:K2=0:K3=0:K8(1)=1:L9=0:A9=0:E9=1:S9=0
140 FOR I=E9TO G9
150 PRINT "YOU ARE ON SOLID",I
160 PRINT HEX(07)
170 PRINT "THE FOLLOWING SOLIDS ARE AVAILABLE"
180 PRINT TAB(4);"RPP(1)   BOX(2)   RAW(3)   ARB8(4)"
190 PRINT TAB(4);"ARB7(5)  ARB6(6)  ARB5(7)  ARB4(8)"
200 PRINT TAB(4);"RCC(9)   TRC(10)  SPH(11)"
210 INPUT "SOLID TYPE IS",T
220 ON TGOTO 380,540,800,240,720,740,760,780,1000,1310,1520
230 PRINT "*** TRY AGAIN ***":GOTO 210
240 REM   *** A R B 8 ***
250 K2=K1+7:IF K2[=120THEN 260:PRINT "NO MORE ROOM FOR ARB8'S":K2=K2-7
:GOTO 1680
260 J=0:FOR K=K1TO K2:J=J+1:PRINT "POINT",J
270 INPUT "X,Y,Z=",A(K),H(K),B(K):NEXT K
280 INPUT "INPUT OK  YES=1 NO=0",P
290 IF P=1THEN 300:GOTO 260
300 PRINT "*** THINKING...THINKING ***"
310 FOR K=K1TO K2
320 C(K)=(H(K)*COS(A))+(A(K)*SIN(A))
330 D(K)=(B(K)*COS(E))+(A(K)*COS(A)*SIN(E))-(H(K)*SIN(A)*SIN(E))
340 IF K[]1THEN 350:IF X8[]X9THEN 350:X8=C(K):X9=C(K):Z8=D(K):Z9=D(K):
GOTO 360
350 GOSUB '07(C(K),D(K))
360 NEXT K
370 K1=K2+1:K3=K2:J=I-L9-S9+1:K8(J)=K1:GOTO 1680
380 REM    ***** R P P *****
390 K2=K1+7:IF K2[=120THEN 400:PRINT "NO MORE ROOM FOR RPP'S":K2=K2-7:
GOTO 1680
400 INPUT "XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX=",R1,R2,R3,R4,R5,R6
410 IF R1]=R2THEN 420:IF R3]=R4THEN 430:IF R5]=R6THEN 440:GOTO 450
420 PRINT "*** ERROR  XMIN ]= XMAX ***":GOTO 400
430 PRINT "*** ERROR  YMIN ]= YMAX ***":GOTO 400
440 PRINT "*** ERROR  ZMIN ]= ZMAX ***":GOTO 400
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
450 INPUT "INPUT OK  YES=1 NO=0",P
460 IF P=1THEN 470:GOTO 400
470 PRINT "*** THINKING...THINKING ***"
480 FOR K=1TO 4:F(K)=R1:J=K+4:F(J)=R2:NEXT K
490 FOR K=1TO 2:G(K)=R3:J=K+4:G(J)=R3:J=K+2:G(K+2)=R4:J=K+6:G(J)=R4:NE
XT K
500 M(1)=R5:M(2)=R6:M(3)=R6:M(4)=R5:M(5)=R5:M(6)=R6:M(7)=R6:M(8)=R5
510 FOR K=K1TO K2:J=K-K3
520 A(K)=F(J):N(K)=G(J):B(K)=M(J):NEXT K
530 GOTO 310
540 REM    ***  B O X ***
550 K2=K1+7:IF K2[=120THEN 560:PRINT "NO MORE ROOM FOR BOX'S":K2=K2-7:
GOTO 1680
560 INPUT "X,Y,Z OF VERTEX=",F(1),G(1),M(1)
570 INPUT "HEIGHT VECTOR=",R1,R2,R3:GOSUB '05(R1,R2,R3)
580 IF C]OTHEN 570
590 INPUT "WIDTH VECTOR=",R4,R5,R6:GOSUB '05(R4,R5,R6)
600 IF C]OTHEN 590
610 INPUT "DEPTH VECTOR=",R7,R8,R9:GOSUB '05(R7,R8,R9)
620 IF C]OTHEN 610
630 INPUT "INPUT OK  YES=1 NO=0",P
640 IF P=1THEN 650:GOTO 560
650 PRINT "*** THINKING...THINKING ***"
660 F(2)=F(1)+R1:G(2)=G(1)+R2:M(2)=M(1)+R3
670 F(3)=F(2)+R4:G(3)=G(2)+R5:M(3)=M(2)+R6
680 F(4)=F(1)+R4:G(4)=G(1)+R5:M(4)=M(1)+R6
690 FOR K=5TO 8:J=K-4
700 F(K)=F(J)+R7:G(K)=G(J)+R8:M(K)=M(J)+R9:NEXT K
710 GOTO 510
720 REM    ***   ARB7   ***
730 K2=K1+6:IF K2[=120THEN 260:K2=K2-6:PRINT "NO MORE ROOM FOR ARB7'S"
:GOTO 1680
740 REM    ***   ARB6   ***
750 K2=K1+5:IF K2[=120THEN 260:K2=K2-5:PRINT "NO MORE ROOM FOR ARB6'S"
:GOTO 1680
760 REM    ***   ARB5   ***
770 K2=K1+4:IF K2[=120THEN 260:K2=K2-4:PRINT "NO MORE ROOM FOR ARB5'S"
:GOTO 1680
780 REM    ***   ARB4   ***
790 K2=K1+3:IF K2[=120THEN 260:K2=K2-3:PRINT "NO MORE ROOM FOR ARB4'S"
:GOTO 1680
800 REM    ***   RAW   ***
810 K2=K1+5:IF K2[=120THEN 820:K2=K2-5:PRINT "NO MORE ROOM FOR RAW'S":
GOTO 1680
820 INPUT "X,Y,Z OF VERTEX=",F(3),G(3),M(3)
830 INPUT "HEIGHT VECTOR=",R1,R2,R3:GOSUB '05(R1,R2,R3)
840 IF C]OTHEN 830
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
850 INPUT "WIDTH VECTOR=",R4,R5,R6:GOSUB '05(R4,R5,R6)
860 IF C]0THEN 850
870 INPUT "DEPTH VECTOR=",R7,R8,R9:GOSUB '05(R7,R8,R9)
880 IF C]0THEN 870
890 INPUT "INPUT OK  YES=1 NO=0",P
900 IF P=1THEN 910:GOTO 820
910 PRINT "*** THINKING...THINKING ***"
920 F(4)=F(3)+R1:G(4)=G(3)+R2:M(4)=M(3)+R3
930 F(6)=F(3)+R4:G(6)=G(3)+R5:M(6)=M(3)+R6
940 F(1)=F(4)+R7:G(1)=G(4)+R8:M(1)=M(4)+R9
950 F(2)=F(3)+R7:G(2)=G(3)+R8:M(2)=M(3)+R9
960 F(5)=F(6)+R7:G(5)=G(6)+R8:M(5)=M(6)+R9
970 FOR K=K1TO K2:J=K-K3
980 A(K)=F(J):H(K)=G(J):B(K)=M(J):NEXT K
990 GOTO 310
1000 REM    ***   RCC   ***
1010 L9=L9+1:IF L9[=5THEN 1020:L9=L9-1:PRINT "NUM RCC'S AND TRC'S ] 5"
:GOTO 1000
1020 INPUT "X,Y,Z OF VERTEX=",V3,V4,V5
1030 INPUT "HEIGHT VECTOR=",H1,H2,H3:GOSUB '05(H1,H2,H3)
1040 IF C]0THEN 1030
1050 INPUT "RADIUS=",H4:N=15:GOSUB '06(H4)
1060 IF C]0THEN 1050
1070 INPUT "INPUT OK  YES=1 NO=0",P
1080 IF P=1THEN 1090:GOTO 1020
1090 PRINT "*** THINKING...THINKING ***"
1100 PRINT "RCC'S AND TRC'S TAKE A WHILE"
1110 L5=((L9-1)*720/N):L6=L5+(360/N)+1:H5=SQR(H1!2+H2!2+H3!2)
1120 L7=SQR(H1!2+H2!2)
1130 FOR K=1TO 360/N:N(K+L5)=0:J=K+L5:O(J)=H4*COS(N*(K-1)):P(J)=H4*SIN
(N*(K-1))
1140 S1=N(J):S2=O(J):S3=P(J)
1150 IF L7[]0THEN 1160:N(J)=-SGN(H3)*S3+V3:O(J)=S2+V4:P(J)=V5:GOTO 1190
1160 N(J)=S1*H1/H5-S2*H2/L7-S3*H1*H3/(L7*H5)+V3
1170 O(J)=S1*H2/H5+S2*H1/L7-S3*H2*H3/(L7*H5)+V4
1180 P(J)=S1*H3/H5+S3*L7/H5+V5
1190 Q(J)=(O(J)*COS(A))+(N(J)*SIN(A))
1200 R(J)=(P(J)*COS(E))+(N(J)*COS(A)*SIN(E))-(O(J)*SIN(A)*SIN(E))
1210 IF T=1OTHEN 1230
1220 O2=J+(360/N):N(O2)=N(J)+H1:O(O2)=O(J)+H2:P(O2)=P(J)+H3
1230 IF J[]1THEN 1240:IF X8[]X9THEN 1240:X8=Q(J):X9=X8:Z8=R(J):Z9=Z8:G
OTO 1250
1240 GOSUB '07(Q(J),R(J))
1250 NEXT K:IF T=1OTHEN 1430
1260 J1=L6:J2=L9*720/N:FOR K=J1TO J2
1270 Q(K)=(O(K)*COS(A))+(N(K)*SIN(A))
1280 R(K)=(P(K)*COS(E))+(N(K)*COS(A)*SIN(E))-(O(K)*SIN(A)*SIN(E))
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
1290 GOSUB '07(Q(K),R(K))
1300 NEXT K:GOTO 1680
1310 REM    ***   TRC   ***
1320 L9=L9+1:IF L9[=5THEN 1330:L9=L9-1:PRINT "NUM RCC'S AND TRC'S ] 5"
:GOTO 1680
1330 INPUT "X,Y,Z OF VERTEX=",V3,V4,V5
1340 INPUT "HEIGHT VECTOR=",H1,H2,H3:GOSUB '05(H1,H2,H3)
1350 IF C]0THEN 1340
1360 INPUT "RADIUS OF BASE=",H4:GOSUB '06(H4)
1370 IF C]0THEN 1360
1380 INPUT "RADIUS OF TOP=",H6:H=15:GOSUB '06(H6)
1390 IF C]0THEN 1330
1400 INPUT "INPUT OK  YES=1 NO=0",P
1410 IF P=1THEN 1420:GOTO 1330
1420 GOTO 1090
1430 IF A9=1THEN 1440:L5=L6-1:L6=L5+25:H4=H6:A9=1:GOTO 1130
1440 A9=0:T4=H1:T5=H2:T6=H3
1450 H1=(T4*COS(A)*COS(E))-(T5*SIN(A)*COS(E))-(T6*SIN(E))
1460 H2=(T5*COS(A))+(T4*SIN(A))
1470 H3=(T6*COS(E))+(T4*COS(A)*SIN(E))-(T5*SIN(A)*SIN(E))
1480 J1=L5+1:J2=L6-1:FOR K=J1TO J2
1490 U(K)=U(K)+T4:O(K)=O(K)+T5:P(K)=P(K)+T6:Q(K)=Q(K)+H2:R(K)=R(K)+H3
1500 GOSUB '07(Q(K),R(K))
1510 NEXT K:GOTO 1680
1520 REM    ***   SPH   ***
1530 S9=S9+1:IF S9[=5THEN 1550:PRINT "ERROR--NUM SPH'S ] 5"
1540 S9=S9-1:GOTO 1680
1550 INPUT "X,Y,Z CENTER=",X(S9),Y(S9),Z(S9)
1560 INPUT "RADIUS=",S8:GOSUB '06(S8)
1570 IF C]0THEN 1560
1580 INPUT "INPUT OK  YES=1 NO=0",P
1590 IF P=1THEN 1600:GOTO 1550
1600 PRINT "*** THINKING...THINKING ***"
1610 S6=(S9-1)*24:N=15
1620 S3=(Y(S9)*COS(A))+(X(S9)*SIN(A)):S4=(Z(S9)*COS(E))+(X(S9)*COS(A)*
SIN(E))-(Y(S9)*SIN(A)*SIN(E))
1630 FOR K=1TO 360/N:J=K+S6:U(J)=S8*COS(N*(K-1)):V(J)=S8*SIN(N*(K-1))
1640 S(J)=U(J)+S3:T(J)=V(J)+S4
1650 IF J[]1THEN 1660:IF X8[]X9THEN 1660:X8=S(J):X9=S(J):Z8=T(J):Z9=T(
J):GOTO 1670
1660 GOSUB '07(S(J),T(J))
1670 NEXT K
1680 NEXT I
1690 K9=K9-L9-S9
1700 GOSUB '04
1710 PRINT "***** S T A R T    P L O T *****"
1720 IF L9=0THEN 2060
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
1730 PRINT "WHEN READY TO PLOT, KEY CONTINUE":PRINT HEX(07):STOP
1740 GOSUB '18(X8,X9,Z8,Z9):GOSUB '01(A,E):GOSUB '02:GOSUB '03
1750 FOR K=1TO L9:GOSUB '18(X8,X9,Z8,Z9):M1=((K-1)*720/N)+1
1760 M2=M1+(360/N)-1:GOSUB '11(Q(M1),R(M1)):FOR J=M1TO M2
1770 GOSUB '12(Q(J),R(J)):NEXT J:GOSUB '12(Q(M1),R(M1)):M3=M2+1:M4=M2+
(360/N)
1780 GOSUB '11(Q(M3),R(M3)):FOR J=M3TO M4:GOSUB '12(Q(J),R(J)):NEXT J
1790 GOSUB '12(Q(M3),R(M3)):GOSUB '12(Q(M1),R(M1))
1800 J1=M1+(180/N):J2=M3+(180/N):GOSUB '11(Q(J1),R(J1))
1810 GOSUB '12(Q(J2),R(J2))
1820 J1=M1+(90/N):J2=M3+(90/N)
1830 GOSUB '11(Q(J1),R(J1)):GOSUB '12(Q(J2),R(J2))
1840 J1=J1+(180/N):J2=J2+(180/N)
1850 GOSUB '11(Q(J1),R(J1)):GOSUB '12(Q(J2),R(J2))
1860 PLOT [,,R]:NEXT K
1870 IF S9]0THEN 2000:IF K9]0THEN 2300
1880 INPUT "SAME SOLIDS DIFF VIEW   1=YES  0=NO",J
1890 IF J]0THEN 1940
1900 INPUT "DO YOU WISH TO ADD MORE SOLIDS  1=YES 0=NO",J
1910 IF J[=0THEN 10
1920 INPUT "HOW MANY SOLIDS DO YOU WISH TO ADD",J
1930 E9=G9+1:G9=G9+J:K9=G9:GOTO 140
1940 INPUT "NEW AZ, EL =",A,E
1950 PRINT "****** I'M WORKING HARD - YOU JUST RELAX ******"
1960 X8=0:X9=0:Z8=0:Z9=0
1970 M1=L9*720/N:FOR K=1TO M1
1980 Q(K)=(O(K)*COS(A))+(N(K)*SIN(A))
1990 R(K)=(P(K)*COS(E))+(N(K)*COS(A)*SIN(E))-(O(K)*SIN(A)*SIN(E))
2000 IF K[]1THEN 2010:IF X8[]X9THEN 2010:X8=Q(K):X9=X8:Z8=R(K):Z9=Z8:G
OTO 2020
2010 GOSUB '07(Q(K),R(K))
2020 NEXT K
2030 PRINT "WHEN READY TO PLOT, KEY CONTINUE":PRINT HEX(07):STOP
2040 GOSUB '18(X8,X9,Z8,Z9):GOSUB '01(A,E):GOSUB '02:GOSUB '03
2050 GOTO 1750
2060 IF S9=0THEN 2280:PRINT "WHEN READY TO PLOT, KEY CONTINUE":PRINT H
EX(07):STOP
2070 GOSUB '18(X8,X9,Z8,Z9):GOSUB '01(A,E):GOSUB '02:GOSUB '03
2080 FOR K=1TO S9:GOSUB '18(X8,X9,Z8,Z9):S1=(K-1)*24+1:S2=S1+23
2090 GOSUB '11(S(S1),T(S1)):FOR J=S1TO S2
2100 GOSUB '12(S(J),T(J)):NEXT J:GOSUB '12(S(S1),T(S1))
2110 PLOT [,,R]:NEXT K
2120 IF K9]0THEN 2300
2130 INPUT "SAME SOLIDS DIFF VIEW   1=YES  0=NO",J
2140 IF J]0THEN 2190
2150 INPUT "DO YOU WISH TO ADD MORE SOLIDS  1=YES 0=NO",J
2160 IF J[=0THEN 10
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
2170 INPUT "HOW MANY SOLIDS DO YOU WISH TO ADD",J
2180 E9=G9+1:G9=G9+J:K9=G9:GOTO 140
2190 INPUT "NEW AZ, EL =",A,E
2200 PRINT "****** I'M WORKING HARD - YOU JUST RELAX ******"
2210 X8=0:X9=0:Z8=0:Z9=0
2220 FOR K=1TO S9:S3=(Y(K)*COS(A))+(X(K)*SIN(A))
2230 S4=(Z(K)*COS(E))+(X(K)*COS(A)*SIN(E))-(Y(K)*SIN(A)*SIN(E))
2240 S6=(K-1)*24+1:S7=S6+23
2250 FOR J=S6TO S7:S(J)=U(J)+S3:T(J)=V(J)+S4
2260 GOSUB '07(S(J),T(J))
2270 NEXT J:NEXT K:IF L9]0THEN 1970:GOTO 2060
2280 PRINT "WHEN READY TO PLOT, KEY CONTINUE":PRINT HEX(07):STOP
2290 GOSUB '18(X8,X9,Z8,Z9):GOSUB '01(A,E):GOSUB '02:GOSUB '03
2300 FOR I=1TO K9
2310 J=K8(I+1)-K8(I):IF J=8THEN 2340:IF J=7THEN 2610
2320 IF J=6THEN 2520:IF J=5THEN 2720:IF J=4THEN 2800
2330 PRINT "ERROR--CANNOT PLOT":STOP :GOTO 10
2340 GOSUB '18(X8,X9,Z8,Z9)
2350 M1=K8(I)
2360 GOSUB '11(C(M1),D(M1))
2370 FOR J=M1TO M1+3:GOSUB '12(C(J),D(J)):NEXT J
2380 GOSUB '11(C(M1+4),D(M1+4))
2390 FOR J=M1+4TO M1+7:GOSUB '12(C(J),D(J)):NEXT J
2400 GOSUB '11(C(M1),D(M1))
2410 GOSUB '12(C(M1),D(M1))
2420 GOSUB '12(C(M1+3),D(M1+3))
2430 GOSUB '12(C(M1+7),D(M1+7))
2440 GOSUB '12(C(M1+4),D(M1+4))
2450 GOSUB '12(C(M1),D(M1))
2460 GOSUB '11(C(M1+1),D(M1+1)):GOSUB '12(C(M1+1),D(M1+1))
2470 GOSUB '12(C(M1+5),D(M1+5))
2480 GOSUB '11(C(M1+6),D(M1+6))
2490 GOSUB '12(C(M1+6),D(M1+6)):GOSUB '12(C(M1+2),D(M1+2))
2500 PLOT [,,R]
2510 GOTO 2860
2520 M1=K8(I):GOSUB '18(X8,X9,Z8,Z9)
2530 GOSUB '11(C(M1),D(M1))
2540 FOR J=M1TO M1+3:GOSUB '12(C(J),D(J)):NEXT J
2550 GOSUB '12(C(M1),D(M1)):GOSUB '12(C(M1+4),D(M1+4))
2560 GOSUB '12(C(M1+5),D(M1+5)):GOSUB '12(C(M1+3),D(M1+3))
2570 GOSUB '11(C(M1+2),D(M1+2)):GOSUB '12(C(M1+5),D(M1+5))
2580 GOSUB '11(C(M1+1),D(M1+1)):GOSUB '12(C(M1+4),D(M1+4))
2590 PLOT [,,R]
2600 GOTO 2860
2610 M1=K8(I):GOSUB '18(X8,X9,Z8,Z9)
2620 GOSUB '11(C(M1),D(M1))
2630 FOR J=M1TO M1+3:GOSUB '12(C(J),D(J)):NEXT J
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
2640 GOSUB '11(C(M1+4),D(M1+4))
2650 FOR J=M1+4TO M1+6:GOSUB '12(C(J),D(J)):NEXT J
2660 GOSUB '12(C(M1+4),D(M1+4)):GOSUB '12(C(M1),D(M1))
2670 GOSUB '12(C(M1+3),D(M1+3)):GOSUB '12(C(M1+4),D(M1+4))
2680 GOSUB '11(C(M1+6),D(M1+6)):GOSUB '12(C(M1+2),D(M1+2))
2690 GOSUB '11(C(M1+1),D(M1+1)):GOSUB '12(C(M1+5),D(M1+5))
2700 PLOT [,,R]
2710 GOTO 2860
2720 M1=K8(I):GOSUB '18(X8,X9,Z8,Z9)
2730 GOSUB '11(C(M1),D(M1))
2740 FOR J=M1TO M1+4:GOSUB '12(C(J),D(J)):NEXT J
2750 GOSUB '12(C(M1),D(M1)):GOSUB '12(C(M1+3),D(M1+3))
2760 GOSUB '11(C(M1+4),D(M1+4)):GOSUB '12(C(M1+1),D(M1+1))
2770 GOSUB '11(C(M1+2),D(M1+2)):GOSUB '12(C(M1+4),D(M1+4))
2780 PLOT [,,R]
2790 GOTO 2860
2800 M1=K8(I):GOSUB '18(X8,X9,Z8,Z9)
2810 GOSUB '11(C(M1),D(M1))
2820 FOR J=M1TO M1+3:GOSUB '12(C(J),D(J)):NEXT J
2830 GOSUB '12(C(M1),D(M1)):GOSUB '12(C(M1+2),D(M1+2))
2840 GOSUB '11(C(M1+3),D(M1+3)):GOSUB '12(C(M1+1),D(M1+1))
2850 PLOT [,,R]
2860 NEXT I
2870 INPUT "SAME SOLIDS DIFF VIEW  1=YES  0=NO",J
2880 IF J]0THEN 2930
2890 INPUT "DO YOU WISH TO ADD MORE SOLIDS  1=YES 0=NO",J
2900 IF J[=0THEN 10
2910 INPUT "HOW MANY SOLIDS DO YOU WISH TO ADD",J
2920 E9=G9+1:G9=G9+J:K9=G9:GOTO 140
2930 INPUT "NEW AZ, EL =",A,E
2940 PRINT "****** I'M WORKING HARD - YOU JUST RELAX ******"
2950 X8=0:X9=0:Z8=0:Z9=0
2960 K2=K8(K9+1)-1: FOR K=1TO K2
2970 C(K)=(H(K)*COS(A))+(A(K)*SIN(A))
2980 D(K)=(B(K)*COS(E))+(A(K)*COS(A)*SIN(E))-(H(K)*SIN(A)*SIN(E))
2990 IF K[]1THEN 3000:X8=C(1):X9=X8:Z8=D(1):Z9=Z8:GOTO 3010
3000 GOSUB '07(C(K),D(K))
3010 NEXT K
3020 IF S9]0THEN 2220:IF L9]0THEN 1970
3030 GOTO 2280
3040 DEFFN'18(U1,U2,V1,V2)
3050 F1=900/(U2-U1)
3060 F2=900/(V2-V1)
3070 IF F1[F2THEN 3090
3080 F1=F2:GOTO 3100
3090 F2=F1
3100 SELECT PLOT 414
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
3110 PLOT [,,R],[100,100,U]
3120 GOSUB '10(U1,V1)
3130 RETURN
3140 DEFFN'10(X,Y)
3150 X1=INT(F1*X-X0+.5)
3160 Y1=INT(F2*Y-Y0+.5)
3170 X0=X0+X1
3180 Y0=Y0+Y1
3190 RETURN
3200 DEFFN'11(X,Y)
3210 GOSUB '10(X,Y)
3220 PLOT [X1,Y1,U]
3230 RETURN
3240 DEFFN'12(X,Y)
3250 GOSUB '10(X,Y)
3260 PLOT [X1,Y1,D]
3270 RETURN
3280 DEFFN'01(A,E)
3290 SELECT PRINT 414
3300 PLOT [,,R]
3310 PLOT [2,,C],[28,,S],[320,20,U]
3320 PRINTUSING 3330,A,E;
3330 %A=-##.#  E=-##.#
3340 SELECT PRINT 005(64)
3350 RETURN
3360 DEFFN'02
3370 PLOT [,,R]
3380 SELECT PRINT 414
3390 PLOT [100,70,U],[0,5,D],[150,0,D],[0,-5,D]
3400 PLOT [1,,C],[14,,S],[,,R],[95,50,U]
3410 PRINTUSING 3420,0;
3420 %#
3430 PLOT [1,,C],[14,,S],[115,0,U]
3440 PRINTUSING 3450,150/F1;
3450 %###.#
3460 PLOT [,,R]
3470 SELECT PRINT 005(64)
3480 RETURN
3490 DEFFN'03
3500 PLOT [,,R]
3510 PLOT [0,999,0],[,,R],[999,0,D]
3520 RETURN
3530 DEFFN'04
3540 F=-10
3550 INPUT "AXES LENGTHS ADJUSTED   1=YES  0=NO",B
3560 IF B]=1THEN 3620:F=F+10
3570 GOSUB '13(X0,X9,Z0,Z9)
```

Table A-XVI. PLOTSOL Program Listing (Continued)

```
3580 PLOT [,,R],[F,F,U],[500,0,D],[,,R]
3590 PLOT [F,F,U],[0,500,D],[,,R]
3600 PRINT "IF THESE LINES ARE OF EQUAL LENGTH THEN SCALE IS OKAY"
3610 PRINT "IF NOT THEN READJUST WITH THE SCALE ADJUST BUTTON":GOTO 35
50
3620 RETURN
3630 DEFFN'05(G,B,D)
3640 C=0:IF G12+B12+D12]0THEN 3660
3650 C=1:PRINT "*** ZERO VECTOR  TRY AGAIN ***":PRINT HEX(07)
3660 RETURN
3670 DEFFN'06(R)
3680 C=0:IF R]0THEN 3700
3690 C=1:PRINT "*** RADIUS [= 0  TRY AGAIN ***":PRINT HEX(07)
3700 RETURN
3710 DEFFN'07(C,D)
3720 IF C]X9THEN 3740:IF C[X8THEN 3730:GOTO 3750
3730 X8=C:GOTO 3750
3740 X9=C
3750 IF D[Z8THEN 3760:IF D]Z9THEN 3770:GOTO 3780
3760 Z8=D:GOTO 3780
3770 Z9=D
3780 RETURN
```

Table A-XVII. DARBIN Program Listing

```
10 DIM X(8),Y(8),Z(8),F(4,6),T(6),A(6),B(6),C(6),R(3,3),F1(6),
     X1(8),Y1(8),Z1(8)
20 %J=##
30 %FACE ####
40 %-#####.####
50 %FOUR POINTS NOT CO-PLANAR IN FACE # # # #  DN=-####.####
60 %#####
70 C9=.00005
80 INPUT "NO. OF VERTICES",V:INPUT "NO. OF FACES",H
90 M=0:N=0:O=0:SELECT PRINT 005(64)
100 FOR J=1TO V:PRINTUSING 20,J
110 INPUT "X(J)",X(J):INPUT "Y(J)",Y(J):INPUT "Z(J)",Z(J)
120 M=M+X(J):N=N+Y(J):O=O+Z(J):X1(J)=X(J):Y1(J)=Y(J):Z1(J)=Z(J)
130 IF X(J)[.00001 THEN 140:X(J)=.00001
140 IF Y(J)[.00001 THEN 150:Y(J)=.00001
150 IF Z(J)[.00001 THEN 160:Z(J)=.00001
160 NEXT J:SELECT PRINT 215(80):PRINT :PRINT "INPUT":GOSUB 470
170 FOR J=1TO H:G=0
180 INPUT "FACE NUMBERS",G:IF G=0 THEN 210:F1(J)=G
190 FOR K=1TO 4:P=10!(4-K)
200 F(K,J)=INT(G/P):G=G-F(K,J)*P:NEXT K
210 PRINTUSING 60,F1(J);:NEXT J:PRINT :E=0
220 FOR J=1TO H:FOR K=1TO 3:L=F(K,J)
230 R(K,1)=X(L):R(K,2)=Y(L):R(K,3)=Z(L)
240 NEXT K:GOSUB 510
250 A(J)=A1:B(J)=B1:C(J)=C1
260 J1=F(4,J):IF J1=0 THEN 300
270 D2=(A(J)*X(J1)+B(J)*Y(J1)+C(J)*Z(J1)-1)/SQR(A(J)!2+B(J)!2
     +C(J)!2)
280 IF ABS(D2)[.001 THEN 300:E=1
290 PRINTUSING 50,F(1,J),F(2,J),F(3,J),F(4,J),D2
300 NEXT J:IF E=0 THEN 310:GOSUB 390:PRINT "SOLUTION":GOSUB 470
310 INPUT "INSIDE ARB (YES=1,NO=0)",J:IF J=0 THEN 80:
    PRINT "THICKNESS ARE"
320 M=M/V:N=N/V:O=O/V:FOR J=1TO H:SELECT PRINT 005(64)
330 PRINTUSING 30,F1(J):INPUT "THICKNESS",T(J)
340 L=1-A(J)*M-B(J)*N-C(J)*O:T1=T(J)
350 IF L[0 THEN 360:T1=-T1
360 E=1/(1-T1*SQR(A(J)!2+B(J)!2+C(J)!2))
370 A(J)=A(J)*E:B(J)=B(J)*E:C(J)=C(J)*E
380 SELECT PRINT 215(80):PRINTUSING 40,T(J);:NEXT J:PRINT :
    GOSUB 390:PRINT "INSIDE ARB":GOSUB 470:GOTO 80
390 FOR J=1TO V:P=0
400 FOR K=1TO H:FOR L=1TO 4
410 IF F(L,K)=J THEN 420:NEXT L:GOTO 440
420 P=P+1:R(P,1)=A(K):R(P,2)=B(K):R(P,3)=C(K)
430 IF P=3 THEN 450
```

Table A-XVII. DARBIN Program Listing (Continued)

```
440 NEXT K:GOTO 460
450 GOSUB 510:X1(J)=A1+SGN(A1)*C9:Y1(J)=B1+SGN(B1)*C9
    :Z1(J)=C1+SGN(C1)*C9
460 NEXT J:RETURN
470 FOR J=1TO V STEP 2:IF J=V THEN 490:K=J+1
480 PRINTUSING 40,X1(J);Y1(J);Z1(J);X1(K);Y1(K);Z1(K):GOTO 500
490 PRINTUSING 40,X1(J);Y1(J);Z1(J)
500 NEXT J:RETURN
510 D1=R(1,1)*R(2,2)*R(3,3)+R(3,1)*R(1,2)*R(2,3)+R(2,1)*R(3,2)*
       R(1,3)-R(3,1)*R(2,2)*R(1,3)-R(2,1)*R(1,2)*R(3,3)-R(1,1)*
       R(3,2)*R(2,3)
520 IF D1[]0 THEN 530:D1=.00001
530 A1=(R(1,2)*(R(2,3)-R(3,3))+R(2,2)*(R(3,3)-R(1,3))+R(3,2)*
       (R(1,3)-R(2,3)))/D1
540 B1=(R(1,3)*(R(2,1)-R(3,1))+R(2,3)*(R(3,1)-R(1,1))+R(3,3)*
       (R(1,1)-R(2,1)))/D1
550 C1=(R(1,1)*(R(2,2)-R(3,2))+R(2,1)*(R(3,2)-R(1,2))+R(3,1)*
       (R(1,2)-R(2,2)))/D1
560 RETURN
570 END
```

Table A-XVIII. BOXIN Program Listing

```
10 DIM P(3),V(3,3),T(6),U(3),W(3,3),M(3),A(8,3),P1(3),V1(3,3),
     B$(3)
20 %######## AND ###### VECTORS ARE NOT NORMAL, ANGLE =-###.##
30 %-####.####
40 SELECT PRINT 215(64):SELECT O :B$(1)="LENGTH":
   B$(2)="WIDTH ":B$(3)="HEIGHT"
50 INPUT "VERTICE",P(1),P(2),P(3)
60 INPUT "LENGTH VECTOR",V(1,1),V(1,2),V(1,3)
70 INPUT "WIDTH VECTOR",V(2,1),V(2,2),V(2,3)
80 INPUT "HEIGHT VECTOR",V(3,1),V(3,2),V(3,3)
90 PRINT :PRINT "INPUT":GOSUB 300
100 FOR J=1TO 3:FOR K=1TO 3:U(K)=V(J,K):NEXT K
110 GOSUB 310:M(J)=M1:FOR K=1TO 3:W(J,K)=U(K):NEXT K:NEXT J
120 E=0:K=1:L=2:N=3:GOSUB 330
130 K=3:L=1:N=2:GOSUB 330
140 K=2:L=3:N=1:GOSUB 330
150 IF E=0 THEN 220
160 FOR J=1TO 3:A(1,J)=P(J):A(2,J)=P(J)+V(1,J):
    A(3,J)=A(2,J)+V(2,J):A(4,J)=P(J)+V(2,J)
170 FOR K=1TO 4:L=K+4:A(L,J)=A(K,J)+V(3,J):NEXT K:NEXT J
180 PRINT "ARB SOLUTION":FOR J=1TO 8 STEP 2:K=J+1
190 PRINTUSING 30,A(J,1);A(J,2);A(J,3);A(K,1);A(K,2);A(K,3):
    NEXT J
200 FOR K=1TO 3:FOR J=1TO 3:V(K,J)=W(K,J)*M1(K):NEXT J:NEXT K
210 PRINT "BOX SOLUTION":GOSUB 300
220 INPUT "INSIDE BOX (YES=1,NO=0)",I:IF I=0 THEN 50
230 INPUT "THICKNESSES",T(1),T(2),T(3),T(4),T(5),T(6)
240 PRINT "THICKNESS ARE":
    PRINTUSING 30,T(1);T(2);T(3);T(4);T(5);T(6)
250 FOR J=1TO 3:P1(J)=P(J)+W(1,J)*T(1)+W(2,J)*T(2)+W(3,J)*T(3)
260 FOR K=1TO 3:L=K+3:V1(K,J)=V(K,J)-W(K,J)*(T(K)+T(L))
270 NEXT K:NEXT J
280 PRINT "INSIDE BOX"
290 PRINTUSING 30,P1(1);P1(2);P1(3);V1(1,1);V1(1,2);V1(1,3),
    V1(2,1);V1(2,2);V1(2,3);V1(3,1);V1(3,2);V1(3,3):GOTO 50
300 PRINTUSING 30,P(1);P(2);P(3);V(1,1);V(1,2);V(1,3),V(2,1);
    V(2,2);V(2,3);V(3,1);V(3,2);V(3,3):RETURN
310 M1=SQR(U(1)!2+U(2)!2+U(3)!2)
320 U(1)=U(1)/M1:U(2)=U(2)/M1:U(3)=U(3)/M1:RETURN
330 D=W(K,1)*W(L,1)+W(K,2)*W(L,2)+W(K,3)*W(L,3):
    IF ABS(D)].0037 THEN 340:RETURN
340 U(1)=W(L,2)*W(N,3)-W(L,3)*W(N,2):
    U(2)=W(L,1)*W(N,3)-W(L,3)*W(N,1):
    U(3)=W(L,1)*W(N,2)-W(L,2)*W(N,1)
350 S=1:IF U(1)*W(K,1)+U(2)*W(K,2)+U(3)*W(K,3)]0 THEN 360:S=-1
360 FOR J=1TO 3:W(K,J)=S*U(J):NEXT J:D=ARCCOS(D):
    E=1:PRINTUSING 20,B$(K),B$(L),D:RETURN :END
```

Table A-XIX. RAWIN Program Listing

```
10 DIM P(3),V(3,3),T(5),U(3),W(3,3),N(3),A(6,3),P1(3),V1(3,3),
     B$(3)
20 %###### AND ###### VECTORS ARE NOT NORMAL, ANGLE =-###.##
30 %-####.####
40 SELECT PRINT 215(64):SELECT 0 :B$(1)="HEIGHT":
   B$(2)="WIDTH ":B$(3)="DEPTH "
50 INPUT "VERTICE",P(1),P(2),P(3)
60 INPUT "HEIGHT VECTOR",V(1,1),V(1,2),V(1,3)
70 INPUT "WIDTH VECTOR",V(2,1),V(2,2),V(2,3)
80 INPUT "DEPTH VECTOR",V(3,1),V(3,2),V(3,3)
90 PRINT :PRINT "INPUT":GOSUB 300
100 FOR J=1TO 3:FOR K=1TO 3:U(K)=V(J,K):NEXT K
110 GOSUB 310:N(J)=N1:FOR K=1TO 3:W(J,K)=U(K):NEXT K:NEXT J
120 E=0:K=1:L=2:N=3:GOSUB 330
130 K=3:L=1:N=2:GOSUB 330
140 K=2:L=3:N=1:GOSUB 330
150 IF E=0 THEN 220
160 FOR J=1TO 3:A(1,J)=P(J):A(2,J)=P(J)+V(1,J):
    A(3,J)=A(2,J)+V(3,J)
170 A(4,J)=P(J)+V(3,J):A(5,J)=P(J)+V(2,J):A(6,J)=A(5,J)+V(3,J):
    NEXT J
180 PRINT "ARB SOLUTION":FOR J=1TO 6 STEP 2:K=J+1
190 PRINTUSING 30,A(J,1);A(J,2);A(J,3);A(K,1);A(K,2);A(K,3):
    NEXT J
200 FOR K=1TO 3:FOR J=1TO 3:V(K,J)=W(K,J)*N(K):NEXT J:NEXT K
210 PRINT "RAW SOLUTION":GOSUB 300
220 INPUT "INSIDE RAW (YES=1,NO=0)",I:IF I=0 THEN 50
230 INPUT "THICKNESSES",T(1),T(2),T(3),T(4),T(5)
240 PRINT "THICKNESS ARE":
    PRINTUSING 30,T(1);T(2);T(3);T(4);T(5)
250 R=(N(1)/N(2))!2:T1=T(5)!2:S9=SGN(T(5)):T3=T(3)+T(4):
    T5=SQR(T1+T1*R)*S9+T(1):T6=SQR(T1+T1/R)*S9+T(2)
260 FOR J=1TO 3:P1(J)=P(J)+W(1,J)*T(1)+W(2,J)*T(2)+W(3,J)*T(3)
270 V1(1,J)=V(1,J)-W(1,J)*T5:V1(2,J)=V(2,J)-W(2,J)*T6:
    V1(3,J)=V(3,J)-W(3,J)*T3:NEXT J
280 PRINT "INSIDE RAW"
290 PRINTUSING 30,P1(1);P1(2);P1(3);V1(1,1);V1(1,2);V1(1,3),
    V1(2,1);V1(2,2);V1(2,3);V1(3,1);V1(3,2);V1(3,3):GOTO 50
300 PRINTUSING 30,P(1);P(2);P(3);V(1,1);V(1,2);V(1,3),V(2,1);
    V(2,2);V(2,3);V(3,1);V(3,2);V(3,3):RETURN
310 N1=SQR(U(1)!2+U(2)!2+U(3)!2)
320 U(1)=U(1)/N1:U(2)=U(2)/N1:U(3)=U(3)/N1:RETURN
330 D=W(K,1)*W(L,1)+W(K,2)*W(L,2)+W(K,3)*W(L,3):
    IF ABS(D)<.9987 THEN 340:RETURN
340 U(1)=W(L,2)*W(N,3)-W(L,3)*W(N,2):
    U(2)=W(L,1)*W(N,3)-W(L,3)*W(N,1):
    U(3)=W(L,1)*W(N,2)-W(L,2)*W(N,1)
```

Table A-XIX. RAWIN Program Listing (Continued)

```
350 S=1:IF U(1)*W(K,1)+U(2)*W(K,2)+U(3)*W(K,3)]0 THEN 360:S=-1
360 FOR J=1TO 3:W(K,J)=S*U(J):NEXT J:D=ARCCOS(D):
    E=1:PRINTUSING 20,B$(K),B$(L),D:RETURN :END
```

Table A-XX. TRCIN Program Listing

```
10 DIM P(3),V(3),T(3),W(3),Q(3),U(3)
20 %-#####.####
30 %WARNING, TOP RADIUS OF INSIDE TRC =-####.##
40 %RADIUS RESET TO .0001 AND THICKNESS(2) = 0
50 SELECT PRINT 215(30)
60 INPUT "VERTICE",P(1),P(2),P(3)
70 INPUT "HEIGHT VECTOR",V(1),V(2),V(3)
80 INPUT "RADIUS OF BASE",R1:INPUT "RADIUS OF TOP",R2
90 INPUT "THICKNESSES",T(1),T(2),T(3):PRINT :PRINT "INPUT"
100 PRINTUSING 20,P(1);P(2);P(3);V(1);V(2);V(3),R1;R2
110 PRINT "THICKNESSES ARE":PRINTUSING 20,T(1);T(2);T(3)
120 H=SQR(V(1)12+V(2)12+V(3)12):G=H/(R1-R2)
130 T2=T(2):F=T(3)/SIN(ARCTAN(G)):S1=R1-F-T(1)/G:S2=R2-F+T(2)/G
140 IF S2]=.0001 THEN 150:PRINTUSING 30,S2:PRINTUSING 40:
    T2=(F-R2)*G:S2=.0001
150 FOR J=1TO 3:U=V(J)/H:Q(J)=P(J)+U*T(1)
160 W(J)=V(J)-U*(T(1)+T2):NEXT J
170 PRINT "INSIDE TRC"
180 PRINTUSING 20,Q(1);Q(2);Q(3);W(1);W(2);W(3),S1;S2
190 GOTO 60:END
```

Table A-XXI. RECIN Program Listing

```
10 DIM P(3),V(3,3),T(3),U(3),W(3,3),M(3),P1(3),V1(3,3),B$(3)
20 %########## AND ########## VECTORS NOT NORMAL, ANGLE=-###.##
30 %-####.####
40 SELECT PRINT 215(64):SELECT D :B$(1)="HEIGHT":
   B$(2)="SEMI-MAJOR":B$(3)="SEMI-MINOR"
50 INPUT "VERTICE",P(1),P(2),P(3)
60 INPUT "HEIGHT VECTOR",V(1,1),V(1,2),V(1,3)
70 INPUT "SEMI-MAJOR VECTOR",V(2,1),V(2,2),V(2,3)
80 INPUT "SEMI-MINOR VECTOR",V(3,1),V(3,2),V(3,3)
90 PRINT :PRINT "INPUT":GOSUB 270
100 FOR J=1TO 3:FOR K=1TO 3:U(K)=V(J,K):NEXT K
110 GOSUB 280:M(J)=M1:FOR K=1TO 3:W(J,K)=U(K):NEXT K:NEXT J
120 E=0:K=1:L=2:M=3:GOSUB 300
130 K=3:L=1:M=2:GOSUB 300
140 K=2:L=3:M=1:GOSUB 300
150 IF E=0 THEN 180
160 FOR K=1TO 3:FOR J=1TO 3:V(K,J)=W(K,J)*M(K):NEXT J:NEXT K
170 PRINT "SOLUTION":GOSUB 270
180 INPUT "INSIDE REC (YES=1,NO=0)",I:IF I=0 THEN 50
190 INPUT "THICKNESSES",T(1),T(2),T(3)
200 PRINT "THICKNESS ARE":
   PRINTUSING 30,T(1);T(2);T(3)
210 T3=T(3):IF M(3)/M(2)1.8 THEN 220:
   T3=T3/(1.016447*(M(3)/M(2))1.071834)
220 FOR J=1TO 3:P1(J)=P(J)+W(1,J)*T(1)
230 V1(1,J)=V(1,J)-W(1,J)*(T(1)+T(2))
240 V1(2,J)=V(2,J)-W(2,J)*T(3):V1(3,J)=V(3,J)-W(3,J)*T3:NEXT J
250 PRINT "INSIDE REC"
260 PRINTUSING 30,P1(1);P1(2);P1(3);V1(1,1);V1(1,2);V1(1,3),
   V1(2,1);V1(2,2);V1(2,3);V1(3,1);V1(3,2);V1(3,3):GOTO 50
270 PRINTUSING 30,P(1);P(2);P(3);V(1,1);V(1,2);V(1,3),V(2,1);
   V(2,2);V(2,3);V(3,1);V(3,2);V(3,3):RETURN
280 M1=SQR(U(1)12+U(2)12+U(3)12)
290 U(1)=U(1)/M1:U(2)=U(2)/M1:U(3)=U(3)/M1:RETURN
300 D=W(K,1)*W(L,1)+W(K,2)*W(L,2)+W(K,3)*W(L,3):
   IF ABS(D)1.0037 THEN 310:RETURN
310 U(1)=W(L,2)*W(M,3)-W(L,3)*W(M,2):
   U(2)=W(L,1)*W(M,3)-W(L,3)*W(M,1):
   U(3)=W(L,1)*W(M,2)-W(L,2)*W(M,1)
320 S=1:IF U(1)*W(K,1)+U(2)*W(K,2)+U(3)*W(K,3)0 THEN 330:S=-1
330 FOR J=1TO 3:W(K,J)=S*U(J):NEXT J:D=ARCCOS(D):
   E=1:PRINTUSING 20,B$(K),B$(L),D:RETURN :END
```

Table A-XXII. TECIN Program Listing

```
10 DIM P(3),V(3,3),T(3),U(3),W(3,3),N(3),P1(3),V1(3,3),Q(3),
     A1(2),A2(2),B1(2),B2(2)
20 %SEMI-MAJOR AND SEMI-MINOR VECTORS NOT NORMAL, ANGLE =-##.##
30 %-####.####
40 %RATIO OF MAJOR AXIS -##.####  RATIO OF MINOR AXIS -##.####
50 %TOP MAJOR AXIS -####.####     TOP MINOR AXIS  -####.####
60 SELECT PRINT 215(00):SELECT D
70 INPUT "VERTICE",P(1),P(2),P(3)
80 INPUT "HEIGHT VECTOR",V(1,1),V(1,2),V(1,3)
90 INPUT "SEMI-MAJOR VECTOR",V(2,1),V(2,2),V(2,3)
100 INPUT "SEMI-MINOR VECTOR",V(3,1),V(3,2),V(3,3):
    INPUT "RATIO",R0
110 PRINT :PRINT "INPUT":GOSUB 460
120 FOR J=1TO 3:FOR K=1TO 3:U(K)=V(J,K):NEXT K
130 GOSUB 470:N(J)=M1:FOR K=1TO 3:W(J,K)=U(K):NEXT K:NEXT J
140 D=W(2,1)*W(3,1)+W(2,2)*W(3,2)+W(2,3)*W(3,3):
    IF ABS(D)[.0007 THEN 260:D=ARCCOS(D):PRINTUSING 20,D
150 J1=2:J2=3:IF N(2)]N(3) THEN 160:J1=3:J2=2
160 C=0:FOR J=1TO 3:IF V(J1,J)=0 THEN 170:C=C+1
170 NEXT J:IF C]1 THEN 180:K=J1:J1=J2:J2=K
180 F=0:FOR J=1TO 3:IF F]ABS(V(J1,J)) THEN 200
190 IF V(J2,J)=0 THEN 200:F=ABS(V(J1,J)):K1=J
200 NEXT J
210 W(J1,K1)=0:F=W(J1,1)+W(J1,2)+W(J1,3)
220 W(J1,K1)=(W(2,1)*W(3,1)+W(2,2)*W(3,2)+W(2,3)*W(3,3))/F
230 FOR J=1TO 3:U(J)=W(J1,J):NEXT J:GOSUB 470
240 FOR J=1TO 3:V(J1,J)=U(J)*M1(J1):W(J1,J)=U(J):NEXT J
250 PRINT "SOLUTION":GOSUB 460
260 INPUT "INSIDE TEC (YES=1,NO=0)",I:IF I=0 THEN 70
270 INPUT "THICKNESSES",T(1),T(2),T(3)
280 PRINT "THICKNESS ARE":PRINTUSING 30,T(1);T(2);T(3)
290 T4=T(3):IF N(3)/N(2)].8 THEN 300:
    T4=T4/(1.016447*(N(3)/N(2))!.071034)
300 Q(1)=W(2,2)*W(3,3)-W(2,3)*W(3,2):
    Q(2)=W(3,1)*W(2,3)-W(2,1)*W(3,3):
    Q(3)=W(2,1)*W(3,2)-W(2,2)*W(3,1)
310 C1=ABS(Q(1)*W(1,1)+Q(2)*W(1,2)+Q(3)*W(1,3))
320 D1=T(1)/C1:D2=T(2)/C1:M=C1*M1(1):N=1
330 FOR K=1TO 2:M4=M1(2)/R0:M5=M1(3)/R0
340 FOR J=1TO 3:M4=M4+M*W(2,J)*V(1,J):M5=M5+M*W(3,J)*V(1,J):
    NEXT J
350 E=M1(2)-M4:F=T(3)*SQR(E!2+M!2)/M:G=F/M
360 A1(K)=M1(2)-F-T(1)*G:A2(K)=M4-F+T(2)*G
370 E=M1(3)-M5:F1=T4*SQR(E!2+M!2)/M:G1=E/M
380 B1(K)=M1(3)-F1-T(1)*G1:B2(K)=M5-F1+T(2)*G1:N=-1:NEXT K
390 A1=.5*(A1(1)+A1(2)):A2=.5*(A2(1)+A2(2)):B1=.5*(B1(1)+B1(2))
    :B2=.5*(B2(1)+B2(2))
```

Table A-XXII. TECIN Program Listing (Continued)

```
400 R2=A1/A2:R3=B1/B2:R1=(R2*B2+R3*A2)/(A2+B2)
410 FOR J=1TO 3:P1(J)=P(J)+W(1,J)*D1+.5*((A1(1)-A1(2))*W(2,J)+
    (B1(1)-B1(2))*W(3,J)):V1(1,J)=V(1,J)-W(1,J)*(D1+D2)
420 V1(2,J)=A1*W(2,J):V1(3,J)=B1*W(3,J):NEXT J
430 PRINT "INSIDE TEC"
440 PRINTUSING 30,P1(1);P1(2);P1(3);V1(1,1);V1(1,2);V1(1,3),
    V1(2,1);V1(2,2);V1(2,3);V1(3,1);V1(3,2);V1(3,3),R1
450 PRINTUSING 40,R2,R3:PRINTUSING 50,A2,B2:GOTO 70
460 PRINTUSING 30,P(1);P(2);P(3);V(1,1);V(1,2);V(1,3),
    V(2,1);V(2,2);V(2,3);V(3,1);V(3,2);V(3,3),R0:RETURN
470 M1=SQR(U(1)12+U(2)12+U(3)12)
480 U(1)=U(1)/M1:U(2)=U(2)/M1:U(3)=U(3)/M1
490 RETURN :END
```

Table A-XXIII. PARB Program Listing

```
10 DIM A(25),B(25),C(25),P(3),A2(25),B2(25),C2(25),R(3,3)
20 %-#####.####
30 % ## ## ##     -#####.#### -#####.#### -#####.####
40 %K=##
50 %INPUT FOR FACE ##
60 %AZIMUTH = -###.##  ELEVATION = -###.##
70 %POINT = -####.#### -####.#### -####.####
80 %PLANE EQ. =-#.##### X+-#.##### Y+-#.##### Z = ##############
.##
90 R1=1E5:SELECT O :SELECT PRINT 215(30):C8=.9999:C9=.00005
100 INPUT "NO. OF FACES (N[26)",F
110 FOR J=1TO F:GOSUB 340:NEXT J
120 PRINT "  PLANES                    POSSIBLE SOLUTION"
130 F2=F-2:F1=F-1:N=0:FOR J=1TO F2:J1=J+1:FOR K=J1TO F1
140 D=A2(J)*A2(K)+B2(J)*B2(K)+C2(J)*C2(K):IF ABS(D)]C8 THEN 240
150 K1=K+1:FOR L=K1TO F
160 D=A2(J)*A2(L)+B2(J)*B2(L)+C2(J)*C2(L):IF ABS(D)]C8 THEN 230
170 D=A2(K)*A2(L)+B2(K)*B2(L)+C2(K)*C2(L):IF ABS(D)]C8 THEN 230
180 R(1,1)=A(J):R(1,2)=B(J):R(1,3)=C(J)
190 R(2,1)=A(K):R(2,2)=B(K):R(2,3)=C(K)
200 R(3,1)=A(L):R(3,2)=B(L):R(3,3)=C(L)
210 GOSUB 280:N=N+1:D=SQR(A1↑2+B1↑2+C1↑2)
220 IF D]R1 THEN 230:A1=A1+SGN(A1)*C9:B1=B1+SGN(B1)*C9:
    C1=C1+SGN(C1)*C9:PRINTUSING 30,J,K,L,A1,B1,C1
230 NEXT L
240 NEXT K:NEXT J
250 INPUT "NEW ARB (YES=1)",J:IF J[]0 THEN 100
260 INPUT "FACE NUMBER (NO MORE=0)",J:IF J=0 THEN 120
270 GOSUB 340:GOTO 260
280 D1=R(1,1)*R(2,2)*R(3,3)+R(3,1)*R(1,2)*R(2,3)+R(2,1)*R(3,2)*
       R(1,3)-R(3,1)*R(2,2)*R(1,3)-R(2,1)*R(1,2)*R(3,3)-R(1,1)*
       R(3,2)*R(2,3)
290 IF D1[]0 THEN 300:D1=.00001
300 A1=(R(1,2)*(R(2,3)-R(3,3))+R(2,2)*(R(3,3)-R(1,3))+R(3,2)*
       (R(1,3)-R(2,3)))/D1
310 B1=(R(1,3)*(R(2,1)-R(3,1))+R(2,3)*(R(3,1)-R(1,1))+R(3,3)*
       (R(1,1)-R(2,1)))/D1
320 C1=(R(1,1)*(R(2,2)-R(3,2))+R(2,1)*(R(3,2)-R(1,2))+R(3,1)*
       (R(1,2)-R(2,2)))/D1
330 RETURN
340 INPUT "A,E-PT=0, 3-PTS=1, EQ=2",O:PRINT :PRINTUSING 50,J:
    IF O[]1 THEN 400
350 SELECT PRINT 005(64):FOR K=1TO 3:PRINTUSING 40,K
360 INPUT "X(K)",R(K,1):INPUT "Y(K)",R(K,2):INPUT "Z(K)",R(K,3)
370 FOR L=1TO 3:IF R(K,L)[]0 THEN 380:R(K,L)=.00001
380 NEXT L:NEXT K:GOSUB 280:A(J)=A1:B(J)=B1:C(J)=C1:
    SELECT PRINT 215(30)
```

Table A-XXIII. PARB Program Listing (Continued)

```
390 PRINT "3-POINTS":PRINTUSING 20,R(1,1);R(1,2);R(1,3);R(2,1);
    R(2,2);R(2,3):PRINTUSING 20,R(3,1);R(3,2);R(3,3):GOTO 490
400 IF 0[]2 THEN 410:INPUT "A",A(J):INPUT "B",B(J):
    INPUT "C",C(J):INPUT "D",D:GOTO 470
410 INPUT "AZIMUTH (ROTATION)",G:INPUT "ELEVATION (FALLBACK)",H
420 INPUT "P(X)",P(1):INPUT "P(Y)",P(2):INPUT "P(Z)",P(3)
430 FOR K=1TO 3:IF P(K)[]0 THEN 440:P(K)=.00001
440 NEXT K:A(J)=COS(G)*COS(H):B(J)=SIN(G)*COS(H):C(J)=SIN(H)
450 D=A(J)*P(1)+B(J)*P(2)+C(J)*P(3)
460 PRINTUSING 60,G,H:PRINTUSING 70,P(1),P(2),P(3)
470 IF D[]0 THEN 480:D=.00001
480 A(J)=A(J)/D:B(J)=B(J)/D:C(J)=C(J)/D
490 G=1/SQR(A(J)12+B(J)12+C(J)12):A2(J)=A(J)*G:B2(J)=B(J)*G:
      C2(J)=C(J)*G
500 PRINTUSING 80,A2(J),B2(J),C2(J),G
510 RETURN :END
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Systems Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
5	Commander US Army Air Mobility Research and Development Laboratory ATTN: SAVDL-EU-MOS Mr. E. V. Merritt Fort Eustis, VA 23604
1	Commander US Army Electronics Command ATTN: DRSEL-RD Fort Monmouth, NJ 07703
5	Director Research and Development Tech Support Agency ATTN: DRSEL-GG-EM, C. Goldy US Army Electronics Command Fort Monmouth, NJ 07703
6	Commander US Army Missile Command ATTN: DRSMI-R DRSMI-RDP, J. Brown (5 cys) Redstone Arsenal, AL 35809
6	Commander US Army Tank Automotive Development Command ATTN: DRDTA-RWL DRDTA-REMA (5 cys) Mr. Tasdemiroglu Warren, MI 48090
2	Commander US Army Mobility Equipment Research & Development Command ATTN: Tech Docu Cen, Bldg. 315 DRSME-RZT Fort Belvoir, VA 22060
5	Commander US Army Mobility Equipment Research & Development Command ATTN: STSFB-BS, L. J. Long Fort Belvoir, VA 22060
1	Commander US Army Armament Command Rock Island, IL 61202
5	Commander US Army Watervliet Arsenal ATTN: SARWV-RDD, R.J. Thierry Watervliet, NY 12189
5	Commander US Army Picatinny Arsenal ATTN: SARPA-AD-C, J. Killen Dover, NJ 07801
1	Commander US Army Harry Diamond Labs ATTN: DRXDO-TI 2800 Powder Mill Road Adelphi, MD 20783

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army TRADOC Systems Analysis Activity ATTN: ATAA-SA White Sands Missile Range NM 88002

Aberdeen Proving Ground

Marine Corps Ln Ofc
Dir, USAMSAA
ATTN: DRXSY-D, Dr. Sperrazza
DRXSY-S, Lawrence Bain
DRXSY-S, R. Bailey (5 cys)